REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
Université M'Hamed BOUGARA –BOUMERDES

FACULTE DE TECHNOLOGIE

Département Ingénierie des Systèmes Electriques

Mémoire de Master

Présenté par :

Mlle OUARKOUB Samira

Mlle BOUBEKEUR Nada

Filière : Télécommunications

Spécialité : Réseaux et Télécommunications

THEME

Etude et mise en œuvre d'une solution SDN

Soutenus devant le jury composé de :

RAHMOUNE	Fayçal	Prof	Université de Boumerdès	Président
BELKACEM	Samia	MCA	Université de Boumerdès	Encadreur
BENSALAMA	Lila	Ing	Université de Boumerdès	Co-Encadreur
NEKAA	Messaouda	MCB	Université de Boumerdès	Examinateur
MECHID	Samira	MAA	Université de Boumerdès	Examinateur

2021/2022

Remerciements

Nous remercions Dieu le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de Dr ***Samia BELKACEM*** Maître *de Conférences [A] à la faculté de technologie* ***M'HAMED BOUGARA BOUMERDES****, nous la remercions pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.*

Nos remerciements s'adressent à Madame ***Lila BENSALAMA*** *chef de département gestion et exploitation du coré IP BACKBONE direction agrégation/division réseau coré pour son aide pratique et son soutien moral et ses encouragements, et à toute l'équipe de la société ALGERIE TELECOM pour leur soutien.*

Nos remerciements vont aussi aux membres du jury pour l'honneur d'avoir voulu examiner et évaluer ce travail.

Nos remerciements s'adressent également à notre chef de département ***Ingénierie des systèmes électriques*** *Monsieur* ***Noureddine MESSAOUDI*** *pour sa générosité et la grande patience dont il est su faire preuve malgré sa charge académique et professionnelle, ainsi tous les enseignants qui ont contribué à notre formation.*

Nous tenons à remercier chaleureusement tous nos proches, de près ou de loin qui nous ont apporté leur sollicitude pour accomplir ce travail.

Table des matières

Table des matières

Liste des abréviations

ASP	Application service provider
API	Application Programming Interface
CC	Cloud Computing
CPU	Central Processing Unit
CLI	command line interface
FAH	Fournisseur d'applications hébergées
IP	Internet protocole
IOT	Internet of Things
IT	Information Technology
ID	Identifiants informatiques
Iaas	Infrastructure as a Service
MAC	Media Access Control
NAT	Network address translation
NTT	Nippon Telecom and Telegraph
NBI	Nourth-Bound Interface
OS	Operating System
ONF	Open Network Fondation
OVS	Open Vswitch
OVSDB	Open Vswitch, Data base

PC	Personnalisée Computer
Paas	Platform as a Service
SSH	Secure Shell
QOS	Qualité de service
RAM	Random Access Memory
SNMP	Simple Network Management Protocol
Saas	Software as a Service
SDN	Software-defined networking
SE	Système d'exploitation
SSL	Secure Sockets Layer
SDN	Software-defined networking
SBI	South-Bound Interface
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTL	Time To Live
VM	Virtual Machine
VDI	Virtual Desktop Infrastructure
VPS	Serveur Virtuel Privé
VLAN	Virtual Local Area Network
WAN	Wide Area Network

Liste des figures

Liste des tableaux

Introduction générale

Introduction générale

Avec l'avènement des innovations technologiques récentes telles que la virtualisation des systèmes et le cloud computing, les limites actuelles des architectures réseaux deviennent de plus en plus problématiques pour les opérateurs et les administrateurs réseaux. Auparavant, elles étaient limitées par la capacité et la bande passante. Il était possible de résoudre les problèmesde capacité simplement en augmentant la bande passante disponible ***(Bouida, 2017)***.

Cependant, cette solution ne semble pas absolue car les réseaux IP traditionnels sont complexes et très difficiles à gérer.

Dans ce réseau classique, lorsqu'un paquet arrive sur un port d'un commutateur ou d'un routeur, celui-ci applique les règles de routage ou de commutation qui sont inscrites dans son système d'exploitation pour l'orienter vers sa destination. De ce fait, les paquets ayant la même destination sont acheminés par la même voie. D'une part, complexe à configurer le réseau en fonction de stratégies prédéfinies, et de la reconfigurer à chaque fois que les besoins par défaut changent. Et d'autre part rendait les choses plus difficiles à faire évoluer en raison du fort couplage qui existe entre le plan de contrôle et le plan de données des équipements d'interconnexion existants.

C'est dans ce contexte qu'a émergé l'idée du ***Software-Defined-Networking*** (**SDN**), qui simplifie la gestion des infrastructures de réseaux physiques hautement distribuées avec une programmabilité et un contrôle centralisé. Les services informatiques peuvent gérer le réseau logiciellement et de bout en bout à partir d'un emplacement central et automatiser certains aspectsdes opérations.

Le SDN, sépare les couches de contrôle et de transport des données, accroît la flexibilité, réduit la complexité de la configuration et accélère les délais de mise sur le marché des nouvelles applications. La réactivité accrue face aux problèmes et pannes améliore la disponibilité du réseau. Enfin, la programmabilité permet aux services informatiques d'automatiser les fonctions réseaux deréduire les coûts opérationnels et il accélère également l'expérience utilisateur.

La programmabilité SDN, s'exécute sur un contrôleur qui est chargé de communiquer les configurations nécessaires aux équipements SDN afin de mettre en œuvre les politiques de transfert uniques et flexibles dont les seules limitations sont liées à la capacité du logiciel

faisant fonctionner le contrôleur ***(Mazzi, 2019)***.

Cette perspective d'un renouveau dans la conception même du réseau, nous a motivés à tenter d'étudier les concepts régissant cette nouvelle vision. Par une étude bibliographique approfondie nous présenterons les différentes normes régissant le paradigme SDN ainsi que ses cas d'usage les plus en vue. Nous implémenterons ensuite son architecture dans un environnement virtuel pour répondre aux problématiques suivantes :

- Quels sont les changements majeurs apportés aux tâches d'administration d'un réseau ?

- Comment exploiter les différentes fonctionnalités apportées par les éléments du SDN pour réaliser une solution facilitant l'exploitation des ressources déployées dans le réseau ?

Enfin nous avons réalisé une solution de sécurité Firewall dans un environnement SDN. Ce mémoire sera réparti en 5 chapitres :

Dans le 1er chapitre, nous avons fait une présentation de l'entreprise de notre stage Algérie Télécom.

Dans le 2eme chapitre, nous essayons de donner une vue générale sur les technologies des réseaux.

Dans le 3eme chapitre, nous nous étalerons sur les principes fondamentaux du paradigme SDN.Il nous permettra d'avoir une idée globale sur son fonctionnement.

Dans le 4eme chapitre, nous évoluons les performances de POX et RYU. Pour cela, nous proposerons une topologie généralement utilisée au sein des entreprises. Puis nous expliquerons les configurations à suivre pour son administration. Des tests seront effectués afin de choisir le contrôleur le plus performant entre les deux.

Dans le 5eme chapitre, nous présenterons l'implémentation de la solution de sécurité Firewall dans un environnement SDN, afin de vérifier la fiabilité et l'utilité d'implémentation de réseaux SDN ***(Aichaoui, 2018).***

Nous terminerons ce mémoire par une conclusion générale et une bibliographie.

Chapitre 1

Présentation Algérie télécom

Chapitre 1 présentation Algérie télécom

1.1 Introduction

Cette phase a pour objectif de présenter l'organisme et la structure d'accueil, l'infrastructure d'Algérie Télécom, ainsi que définir le Backbone IP/MPLS,ses services supportés, sur lequel est basé la plateforme SDN d'Algérie Télécom.

1.2 Présentation de l'organisme d'accueil

1.2.1 Présentation d'Algérie Télécom

Algérie Télécom est le leader sur le marché Algérien des télécommunications qui connait une forte croissance. Offrant une gamme complète de services de voix et de données aux clients résidentiels et professionnels.

Algérie Télécom, est une société par actions à capitaux publics opérant sur le marché des réseaux et services de communications électroniques.

Sa naissance a été consacrée par la loi 2000/03 du 5 août 2000, fixant les règles générales relatives à la poste et aux télécommunications ainsi que les résolutions du conseil national aux participations de l'Etats (CNPE) du 1er Mars 2001 portant sur la création d'une Entreprise Publique Economique dénommée « Algérie Télécom ».

Algérie Télécom est donc régie par cette loi qui lui confère le statut d'une entreprise publique économique sous la forme juridique d'une société par actions SPA au capital social de 50.000.000.000 Dinars et inscrite au centre du registre de commerce le 11 mai 2002.

Entrée officiellement en activité à partir du 1er janvier 2003, elle s'engage dans le monde des Technologies de l'Information et de la Communication avec trois objectifs :

- Rentabilité.
- Efficacité.
- Qualité de service.

Son ambition est d'avoir un niveau élevé de performance technique, économique, et sociale pour se maintenir durablement leader dans son domaine, dans un environnement devenu concurrentiel.

Son souci consiste, aussi, à préserver et développer sa dimension internationale et participer à la promotion de la société de l'information en Algérie.

1.2.2 Organisation de l'organisme d'accueil

Algérie Télécom est organisée en trois pôles ; Pôle Marketing et Commerciale, Pôle Administration Générale, Pôle Infrastructure et Réseaux qui regroupent la Division Réseau Core, la Division de Réseau de transport et la Division Réseau d'accès.

On trouve aussi 53 Directions opérationnels des Télécommunications situent dans 48 wilaya en Algérie avec trois au niveau de la wilaya d'Alger, deux au niveau d'Oran et deux au niveau de Constantine. Comme illustré dans la **figure (1.1).**

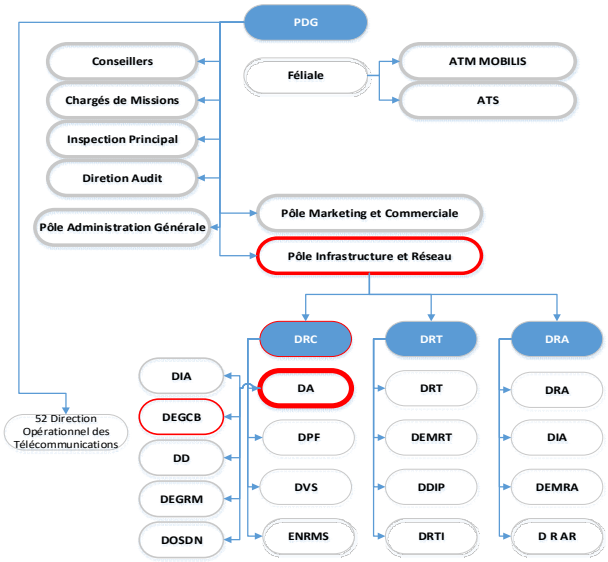


Figure 1.1:Organigramme d'Algérie télécom

Pôle Infrastructure et Réseaux

Division Réseau Core (DRC) : la Division où se trouve le Département d'accueil

- Direction Agrégation (DA)/Département Exploitation et Gestion de Core IP Backbone (DEGCB).
- Direction Plates formes (DPF)
- Direction Voix et Services (DVS)
- Etablissement National RMS Kouba (ENRMS)

Division Réseau Transport (DRT)

- Direction du Réseau de Transport (DRT)
- Direction d'Exploitation et Maintenance du Réseau de Transport (DERT)
- Direction de réseau de Transport International (DRTI)

- Direction Développement de l'Infrastructure Passif (DDIP)

Division Réseau d'Accès (DRA)

- Direction d'Ingénierie d'Accès (DIA)
- Direction du Réseau d'Accès (DRA)
- Direction d'Exploitation et Maintenance du Réseau d'Accès (DEMRA)
- Direction réseau d'Accès Radio (DRAR)

1.3 Infrastructure de réseau d'Algérie Télécom

Algérie Telecom est l'un des importants fournisseurs de services dans le domaine des télécom en Algérie. Il recouvre l'ensemble du territoire national, et chaque wilaya dispose de sa propre infrastructure.

L'infrastructure de réseau d'Algérie télécom de la wilaya d'Alger repose sur un modèle multicouche évolutif traditionnel comme l'illustre la Figure I.2.

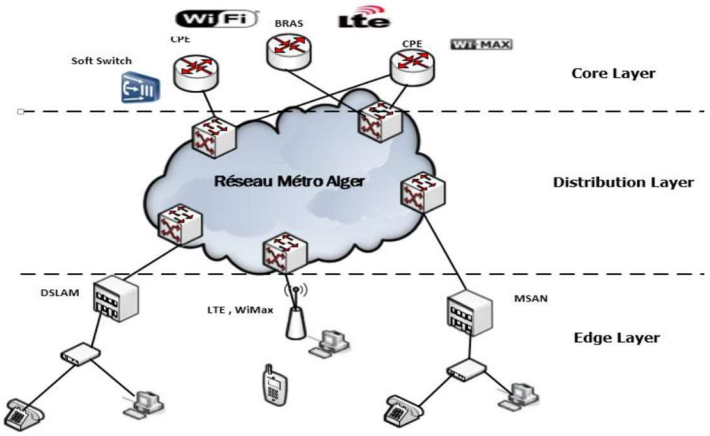


Figure 1.2:Infrastructure de réseau d'Algérie télécom de la wilaya d'Alger

- **Couche Core de réseau (Core Layer)** : contient les plateformes de services détient par Algérie télécom et fournis à la disposition de ces clients.
- **Couche de distribution (Distribution Layer)** : représente le réseau Métro Switch de la wilaya, regroupe les données reçues à partir des nœuds d'accès avant qu'elles ne soient transmises vers la couche cœur de réseau, en vue de leur commutation vers la plateforme appropriée.
- **Couche d'accès (Edge Layer)** : représente l'interface avec les clients de service et assure l'accès au réseau à l'aide des périphériques spécifiés.

1.3.1 Backbone IP/MPLS d'Algérie Télécom

Dans cette partie nous présentons succinctement le backbone IP/MPLS d'Algérie Telecom.

a- Structure du réseau RMS d'Algérie Telecom

Le réseau RMS (Réseau Multiservices) d'Algérie Télécom est un réseau de nouvelle génération (Next Génération Network) et d'envergure nationale, qui dispose d'une architecture de fonctionnement en couche. Ce Backbone détient deux couches principales : Une couche Cœur et une couche Edge, réparties en deux sites (des sites Primaires et des Sites Secondaires).

Le réseau RMS (Réseau Multiservices) est conçu afin de supporter et fédérer tous les types de protocoles, grâce à un ensemble d'équipements réseau comprenant les larges bandes tels que des routeurs (Core et Edge), des Softswitchs, des media Gateway,le tout étant relié par des liaisons en fibre optique. La solution de raccordement des sites sur le backbone RMS permet de garantir et offrir : Le débit, la fiabilité,la qualité de service,la sécurité, la disponibilité, l'extensibilité et la souplesse,la redondance, VOIP (Voice Over IP), Internet, vidéoconférence.

Cette infrastructure fonctionne par l'attribution des Réseaux Privé Virtuels en s'appuyant sur le protocole MPLS.

b- Architecture du backbone RMS d'Algérie Telecom

Le RMS offre à ses clients plusieurs solutions pour l'interconnexion des sites à savoir liaison spécialisée, Wi Max (Worldwid interoperabiklity for microwave access), ADSL (Asymmetric Digital Subscriber Line), MSAN (MultiService Access Node) …, la figure ci-dessous montre le schéma global de l'architecture Backbone IP/MPLS d'Algérie télécom :

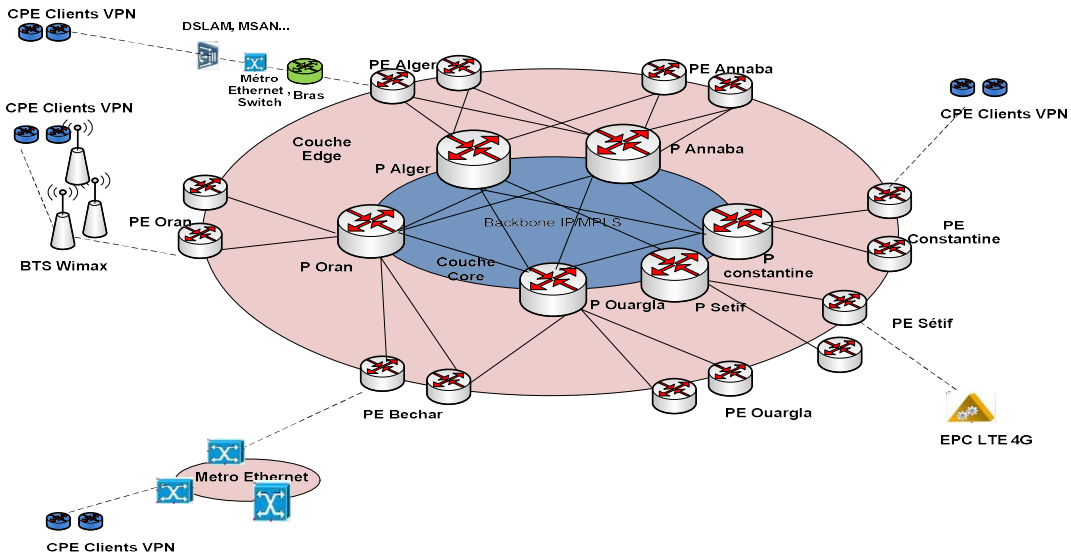


Figure 1.3:Schema global de l'architecture backbone IP/MPLS d'Algérie télécom

c- Les services supportés par le réseau RMS

Ce concept des réseaux permettra à Algérie télécom de mettre en place progressivement des services et des produis à valeur ajoutée qui répondent aux besoins futurs de la clientèle diversifiée (entreprises, institutions, administration, particuliers et entreprise de communication). Les services sont les suivants :

- Les services téléphoniques : tels qu'interconnexion des réseaux mobile et fixe, …
- Les services des données : nous citons les services des données les plus répondus
  - VPN L2 (Virtuel Private Network Layer 2) : ce service est garanti grâce aux circuits virtuels qui donnent aux clients l'impression d'être reliés directement avec une liaison dédiée.
  - VPN L3 (Virtuel Private Network Layer3) : ce service permet de transport fiable de données et une gamme de service à haut qualité (rapidité, fiabilité).
  - Les services vidéo et multimédia (vidéo conférence,)

1.3.2 Concepts liés aux réseaux NGN

Les réseaux traditionnels (GSM/GPRS (Global System for Mobile communication /General Packet Radio Service), RTC/RNIS (Réseau Numérique à l'intégration de Service /Réseau téléphonique commuté), les réseaux de données (FR (Frame Réaly)), ATM (Asynchronous Transfer Mode) ; IP (Internet Protocol), le réseau câblé pour le diffusion T, présentent plusieurs inconvénient telle que l'incompatibilité avec les réseaux de nouveaux services, ce qui a poussé à adopter une nouvelle architecture de réseau NGN.

Le NGN est un réseau multiservices dont le plan de commandes (signalisation, commande) est séparé du plan de transport/commutation. Il peut prendre en charge voix, donnée et vidéo.

Le réseau NGN repose sur une architecture en couches indépendantes communicantes via des interfaces ouvertes et normalisées à savoir :

a) Couche transport

Se divise en deux sous couches

La couche accès : regroupe les fonctions et les équipements permettant de gérer l'accès des utilisateurs au réseau selon la technologie d'accès.

La couche transit : responsable de l'acheminement du trafic voix ou données dans le cœur de réseau IP, selon le protocole utilisé. L'équipement important à ce niveau est le « media Gateway » qui est le médiateur entre les réseau traditionnels TDM (Time Division

Multiplexing) et les nouveaux réseaux multiservices basés sur IP.

b) Couche contrôle

Gère l'ensemble des fonctions de contrôle des services en général et de contrôle d'appel en particulier pour le service voix. À ce niveau, le soft switch est l'équipement utilisé pour le service d'appel.

c) Couche service

Permet la fourniture de services, elle regroupe deux types d'équipement, les serveurs d'application SIP (Session Initiation Protocol) et les « enablers » la figure ci-dessous illustre l'architecture des couches :

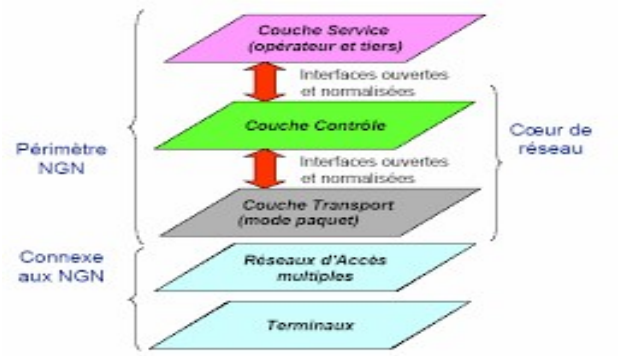


Figure 1.4:Architecture NGN par couche

1.4 Présentation du CONTRÔLEUR NORTHSTAR d'Algérie Télécom

Juniper Networks North Star Controller est une solution d'ingénierie de trafic puissante et flexible qui permet une visibilité et un contrôle granulaires des flux IP/ MPLS dans les grands fournisseurs de services et les réseaux d'entreprise. Il permet aux opérateurs de réseau d'optimiser leur infrastructure de réseau grâce à une surveillance proactive, une planification et un routage explicite des charges de trafic importantes de manière dynamique en fonction des contraintes spécifiées.

Cela a permet à Algérie Télécom de faire fonctionner son réseau, tout en assurant la prévisibilité, la résilience et Path Computation Client (PCC).

Nort Star Controller fournit les fonctions suivantes :

- Offre un calcul et un provisionnement de chemin LSP en temps réel.
- Fournit une optimisation globale des LSP.
- Déclenche le réacheminement du LSP lorsqu'il est nécessaire de réoptimiser le réseau.

- Modifier la bande passante LSP lorsqu'il y a une augmentation de la bande passante d'une application.
- Modifie d'autres attributs LSP sur le routeur, la priorité de configuration et garder la priorité.

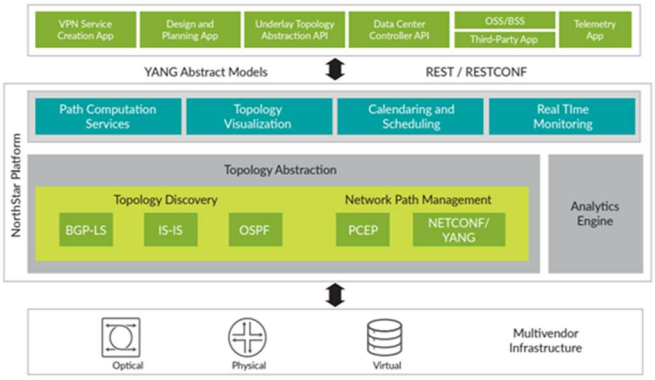


Figure 1.5:Architecture NorthStar

1.4.1 Description du produit

Les fournisseurs de services et les grandes entreprises telles que Algérie Télécom subissent une pression croissante pour déployer rapidement des services tout en réduisant les CapEx et OpEx globaux. Il n'est plus économiquement viable de tolérer l'inactivité de l'infrastructure réseau tout en anticipant passivement les besoins de croissance. Les gestionnaires de réseau d'Algérie Télécom ont passé à un modèle opérationnel où les mises à niveau de capacité sont ciblées, axées sur les services et peuvent s'adapter dynamiquement aux besoins en constante évolution de leurs clients en temps réel.

De plus, NorthStar Controller est capable de découvrir dynamiquement la topologie du réseau en s'appariant via IGP (ISIS-TE, OSPF-TE) et en écoutant les mises à jour BGP-LS, tandis que la découverte de la topologie du réseau optique et des chemins virtuels optiques peut être effectuée via Appels d'API REST utilisant la notion de topologie abstraite telle que définie dans les normes IETF dans le plan optique.

Source Packet Routing in Networks (SPRING) est une méthode de routage compatible SDN où un contrôleur centralisé maintient les ressources du réseau et dirige le trafic en fonction des besoins de l'application.

NorthStar Controller peut également recevoir des données de surveillance des performances en temps réel à partir de l'interface de télémétrie Junos ou d'autres solutions similaires. En diffusant des données vers un système de gestion des performances, les administrateurs réseau peuvent mesurer les tendances d'utilisation des liaisons et des nœuds,

et dépanner des problèmes tels que la congestion du réseau en temps réel.

Du côté de la gestion, NorthStar Controller dispose d'un ensemble d'interfaces nord standard basées sur le Web qui sont RESTful/APIs basées sur Thrift qui facilitent l'intégration avec les systèmes OSS/BSS existants, les scripts tiers de modélisation/planification du trafic ou les applications personnalisées. Ce riche ensemble de commandes permet au contrôleur NorthStar d'interagir avec la plupart des systèmes de gestion de réseau en place.

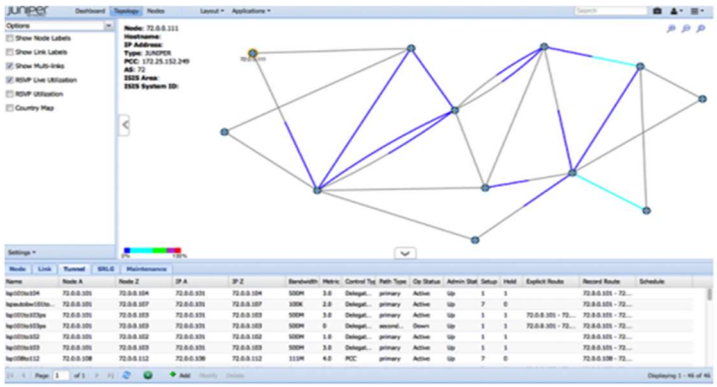


Figure 1.6:Northstar live network visualisation

1.4.2 Intégration optique de paquets multicouches

NorthStar Controller fournit des vues multicouches du réseau en interagissant dynamiquement avec les contrôleurs de transport. AT n'a plus besoin de configurer manuellement les groupes de liaison à risque partagé (SRLG) sur les périphériques réseau pour planifier les LSP. Au lieu de cela, NorthStar Controller, communique directement avec les contrôleurs de transport via REST/RESTCONF pour apprendre la topologie abstraite telle que définie par les normes IETF.

NorthStar Controller a une vue globale de la demande de bande passante dans le réseau et effectue des calculs de chemin externes après avoir interrogé la base de données d'ingénierie du trafic. Il modifie ensuite un ou plusieurs attributs LSP et envoie une mise à jour au client (PCC). Le client utilise les paramètres qu'il reçoit du contrôleur NorthStar pour resignaler le LSP. Cela permet au contrôleur NorthStar de fournir une opération coopérative de fonctionnalité distribuée utilisée pour relever les défis spécifiques d'un calcul de chemin contraint inter domaine le plus court. Il élimine les scénarios de congestion dans lesquels les flux de trafic sont mappés de manière inefficace sur les ressources disponibles, entraînant une sur utilisation de certains sous-ensembles de ressources réseau, tandis que d'autres ressources restent sous-

utilisées.

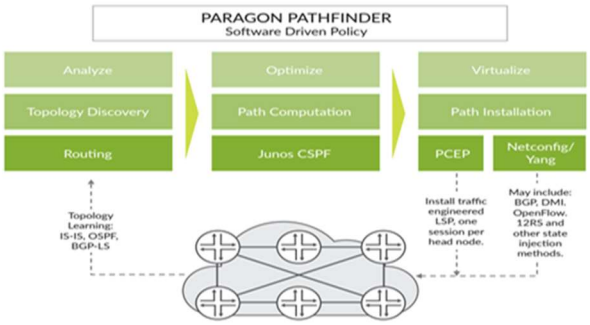


Figure1.7:Typical Northstar Controller workflow

1.4.3 Solution multifournisseur

Le contrôleur NorthStar aidé AT à concevoir des créations de chemins divers (LSP) sans avoir à configurer manuellement SRLG sur les périphériques réseau. NorthStar reçoit également des mises à jour en temps réel des contrôleurs de transport, permettant à AT d'effectuer des tâches d'optimisation telles que la conception et la planification de réseaux multicouches ainsi que la planification coordonnée de la maintenance des paquets et des nœuds optiques.

1.4.4 Avantages

Les principaux avantages de l'architecture d'application du contrôleur NorthStar d'Algérie Télécom sont les suivants :

- Fournit un ordre et une synchronisation spécifiques des chemins signalés sur les éléments de réseau routés.
- Permet une vue globale de l'état du réseau pour la surveillance, gestion et planification proactive.
- Présente un état de réseau prévisible et déterministe avec marge d'erreur de prévision de la demande.
- Offre un état distribué minimisé, une efficacité accrue des éléments de réseau existants via le déchargement du traitement du plan de contrôle.
- Fournit une simplicité opérationnelle grâce à l'activation d'un point de contrôle SDN sur des éléments disparus du réseau.

Conclusion

Algérie télécom s'engage dans le monde des Technologies de l'Information et de la communication avec trois objectifs : Rentabilité, efficacité et qualité de service. Son ambition

est d'avoir un niveau élevé de performance technique, économique et sociale pour se maintenir durablement leader dans son domaine, dans un environnement devenu concurrentiel. Son souci consiste aussi à préserver et développer sa dimension internationale et participer à la promotion de la société de l'information en Algérie.

Parmi les technologies en cours d'étude et de réalisation actuellement à Algérie télécom est l'implémentation du la plateforme SDN, pour répondre à l'évolution des attentes des clients et de fournir de nouveaux services plus

Dans la première partie de notre travail nous avons présenté l'étude actuelle de réseaux

Chapitre 2

Etude actuelle des Réseaux

Chapitre 2 Etude sur les réseaux actuels

2 Introduction

Les réseaux à grande échelle sont coûteux à exploiter et les fournisseurs de services sont toujours à la recherche des moyens pour réduire leur capital et les coûts opérationnels,ces réseaux ont toujours été conçus et construits sous forme de silos technologiques : postes de travail, la virtualisation, serveurs, réseaux, stockage, logiciels et applications. Mais avec toutes ces innovations, nous rencontrons toujours des problèmes potentiellement attaquables et les services virtuels peuvent être mis hors fonction.

Dans ce chapitre, nous allons mis en évidence les réseaux actuels d'Algérie télécom, telque la virtualisation cloud computing et le besoin de réseau programmable.

2.1 Réseau traditionnel

Les réseaux actuels se fondent sur le principe de couplage des deux plans, le plan de contrôle (décide comment gérer le trafic réseau) et le plan de données (transfère le trafic en fonction des décisions prises par le plan de contrôle) qui sont regroupées au niveau de chacun des périphériques réseau. En effet les équipements IP (Internet Protocol) sont à lafois des éléments de contrôle, qui prennent les décisions de routage et des éléments responsables de la transmission des données. Il s'agit d'un obstacle fondamental à l'innovation au niveau des infrastructures réseau.

Afin de comprendre le fonctionnement de cette architecture, il est primordial de comprendre le fonctionnement de ses composants clés. Un routeur ou un commutateur traditionnel a trois plans de fonctionnement architecturaux de base : plan de gestion, plan decontrôle et plan de données.

2.1.1 Plan de gestion

Concerne l'accès administratif à un périphérique réseau. Se connecter au dispositif via une session Telnet ou SSH (Secure Shell), ou disposer d'une station de gestion exécutant leprotocole SNMP (Simple Network Management Protocol) qui communique avec un périphérique réseau.

2.1.2 Plan de Contrôle

C'est l'intelligence du boitier, il lui permet de déterminer les destinations que les paquets doivent atteindre. Il comprend des mécanismes de transmission de couche 2 et 3, notammentla table de voisinage, la table topologique, la table de routage, etc.

2.1.3 Plan de données

C'est la capacité de traitement du boitier, il lui permet de faire transiter les paquets à très grande vitesse vers la destination.

Ces trois blocs (le plan de gestion, le plan de contrôle et le plan de données) sont intégrésdans chaque nœud, comme le montre la ***figure (2.1)***.

Figure 2.1:Architecture des équipements IP traditionnels

Cette architecture présente plusieurs avantages, principalement de minimiser la charge sur l'administrateur réseau. Au lieu d'avoir un seul point d'administration, cette charge sera répartie entre plusieurs sites gérés par plusieurs administrateurs. Malgré les avantages qu'elle présente, elle peut exposer le réseau à des limitations majeures ***(Ccna R/S, 2016).***

2.2 Les limites des réseaux traditionnels

On a pris quelques limites des réseaux traditionnels sont les suivants :

- ***Limites d'automatisation***

Il y a des types des réseaux qui ont toujours des instructions quichangent dynamiquement. Donc à chaque fois il faut que l'administrateur ait accès à chaqueéquipement réseau afin de modifier certaines configurations.

- ***Erreur prône***

Une haute possibilité d'avoir une erreur causée par l'administrateur à cause de la gestion et

la configuration manuelle sur chaque équipement.

- ***Consommation du temps***

Ces réseaux consomment beaucoup du temps de réponse car chaque équipement fait les calculs individuellement.

- ***Complexité***

L'ajout ou la modification d'équipements et l'implémentation des politiques réseaux sont complexes, longues et peuvent être source d'interruptions de service. Ce qui décourage les modifications et l'évolution du réseau.

- ***Passage à l'échelle***

L'impossibilité d'avoir un réseau qui s'adapte au trafic a obligé les opérateurs à sur-provisionner leurs réseaux, ce qui ajoute une complexité de gestion sur le plan de contrôle.

- ***Incompatibilité et dépendance***

Chaque nœud est un boitier propriétaire.

- ***Limites d'innovation***

Tous les équipements des réseaux traditionnels sont des « closed systèmes » donc ils ne supportent pas l'innovation des systèmes, d'où il faut acheter d'autres équipements qui le supportent.

Gérer des machines réseau à une échelle industrielle relève du cauchemar. Ce sont des pièces importantes, spécialisées et onéreuses qui représentent un casse-tête pour les entreprises.

La vie d'un administrateur réseau est un cycle sans fin de réflexion, de travail fastidieux,de configurations, de mises à niveau et de remplacements. Pour les entreprises et les opérateurs, c'est une activité coûteuse.

C'est un univers où prévaut encore la dépendance vis-à-vis du fournisseur, c'est-à-dire qu'une entreprise choisit un fournisseur, tel que Cisco, Huawei ou Juniper, et effectue tous ses achats auprès d'un seul fournisseur ***(OUMEDDI, 2016)***.

2.3 La virtualisation

Il est temps de dépasser la logique traditionnelle et d'adapter une logique qui permettra d'automatiser les fonctionnalités de réseau, afin de concentrer les efforts sur les travaux amélioratifs et innovantes pour le réseau ***(Ccna R/S, 2016).***

2.3.1 La définition

La virtualisation est une technologie qui fait appel aux logiciels pour créer des versions logicielles des systèmes et des services de TI, qui étaient traditionnellement mise en œuvre sur du matériel physique distinct. Ces versions logicielles (ou instances virtuelles) peuvent accroître considérablement l'efficacité des systèmes et réduire les coûts. Nous pouvons utiliser le matériel au maximum de sa capacité en répartissant ses fonctionnalités entre plusieurs services différents.

Avant de mettre en œuvre un logiciel de virtualisation au sein de notre organisation, il est important de comprendre les risques associés à une telle utilisation et d'assurer la protection de notre réseau, de nos systèmes et de notre information.

La virtualisation permet d'exécuter nos applications tout en utilisant moins de serveurs physiques. Les applications et les logiciels s'exécutent virtuellement sur un système informatique simulé que l'on appelle une machine virtuelle (VM pour Virtual Machine).

LaVM possède toutes les caractéristiques d'un serveur informatique, sans qu'il ne soit nécessaire d'y connecter du matériel. La VM est prise en charge par l'hyperviseur.

La virtualisation permet d'utiliser les serveurs matériels au maximum de leurs capacitéset de fournir, à l'hyperviseur, toutes les ressources nécessaires pour la prise en charge des VM.

Aujourd'hui, la virtualisation est une pratique courante dans l'architecture informatique d'entreprise. C'est aussi la technologie qui dynamise l'économie du cloud computing.

La virtualisation permet aux fournisseurs de cloud de servir les utilisateurs avec leur matériel informatique physique existant ; elle permet aux utilisateurs du cloud d'acheter uniquement les ressources informatiques dont ils ont besoin lorsqu'ils en ont besoin, et de faire évoluer ces ressources de manière rentable à mesure que leurs charges de travail augmentent.

La virtualisation accroîtra les performances de l'infrastructure de notre organisation, puisqu'elle lui sera alors possible de faire ce qui suit :

- Exécuter plusieurs systèmes d'exploitation sur une machine physique.
- Répartir les ressources système entre les VM (p. ex. équilibrage de charge).
- Renforcer le contrôle des ressources.
- Créer des Appliance de sécurité virtualisées (p. ex. pare-feu).
- Déplacer, copier et enregistrer facilement les VM sur d'autres systèmes.
- Exécuter une infrastructure de poste de travail virtuel (VDI pour Virtual DesktopInfrastructure) dans les installations et à distance **(Canada, 2020).**

2.3.2 Usages de la virtualisation

Il existe différents types de virtualisation. On dénombre sept principaux domaines de l'informatique où la virtualisation est couramment utilisée :

- Virtualisation de stockage

La virtualisation de stockage est lorsque l'espace de stockage physique, de plusieurs appareils d'un réseau, est unifié au sein d'un appareil de stockage virtuel géré depuisune console centrale. Pour virtualiser le stockage, nous avons besoin d'un logiciel devirtualisation capable d'identifier les capacités disponibles sur les appareils physiques et d'agréger ces capacités au sein d'un environnement virtuel.

Le stockage virtuel ressemble à un disque dur physiquestandard. Le stockage virtuel est un composant clé d'une stratégie IT de type infrastructure hyperconvergée et permet aux administrateurs informatiques de rationaliser les activités destockage comme la sauvegarde, l'archivage et la récupération **(Citrix, 2022)**.

- Virtualisation de réseau

Face à l'utilisation généralisée des environnements virtualisés, de nombreuses entreprises virtualisent également leurs réseaux. La virtualisation réseau consiste à diviser la bande passante disponible en canaux indépendants, chacun étant affecté à un serveur ou un appareil, en fonction des besoins. La virtualisation réseau facilite les tâches de programmation et de provisionnement du réseau, telles que l'équilibrage des charges et la protection par pare-feu, sans avoir à toucher à l'infrastructure sous-jacente. En règlegénérale, les équipes informatiques gèrent les composants logiciels à l'aide d'une console d'administration basée sur un logiciel (également connu sous le nom de réseau logiciel ou SDN).

Une autre méthode est la virtualisation des fonctions réseau, qui consiste à virtualiser les Appliance matérielles offrant des fonctions dédiées pour un réseau (comme l'équilibrage des charges ou l'analyse du trafic), afin de faciliter le provisionnement et la gestion de ces Appliance.

À la mesure que les besoins évoluent, la virtualisation réseau simplifie les modalités de déploiement, de mise à l'échelle et d'ajustement des capacités de calcul pour les équipes informatiques **(Citrix, 2022)**.

- Virtualisation de données

La virtualisation de données permet à une application d'accéder aux données et de les exploiter sans avoir besoin des détails sur l'emplacement physique ou le format de ces données. Nous pouvons ainsi créer une représentation de données à partir de plusieurs sources, sans déplacer ni copier ces données. Cette agrégation des données se fait avec un logiciel de virtualisation de données, qui intègre et visualise virtuellement ces données au travers d'un tableau de bord, permettant aux utilisateurs d'accéder à de grands ensembles dedonnées depuis un point unique, où que les données soient stockées. La virtualisation de données est essentielle à tout type d'application d'analytique ou d'intelligence **(Citrix, 2022)**.

- Virtualisation de postes

La virtualisation de postes nous permettons de simuler une charge de poste de travail afin d'accéder à des postes à distance depuis un appareil connecté, comme un client léger à un bureau. Ces postes virtuels permettent un accès portable plus sécurisé aux ressources du Datacenter **(Citrix, 2022)**.

- La virtualisation hardware ou matérielle

La virtualisation hardware est utilisée pour la gestion des applications. Elle consiste à regrouper tous les serveurs physiques transformés en serveurs virtuels en un seul serveur physique. La virtualisation hardware implique donc directement la gestion du matériel informatique. Elle permet de libérer le processeur pour le faire tourner plus vite et plus efficacement, tout en ne représentant qu'une seule entité pour les autres matériels.

La connexion ou l'accès à la machine virtuelle, via le serveur physique général, puis le serveur virtuel individuel, est rendue possible par le logiciel hyperviseur. Celui-ci est chargéde

faire le lien entre le matériel informatique et l'environnement virtuel. Il répartit la mémoire utile sur chacun des serveurs virtuels pour optimiser leur fonctionnement ***(Erell, 2020)***.

- Virtualisation d'applications

Avec la virtualisation d'applications, les utilisateurs peuvent exécuter des applications sous une forme distincte, indépendamment du système d'exploitation utilisé. Cela permet notamment d'exécuter une application Microsoft Windows sur un système d'exploitation Linux ou Mac **(Citrix, 2022)**.

- La virtualisation serveur

Précédemment évoquée, permet d'exécuter plusieurs systèmes d'exploitation sur un seul serveur physique sous forme de machines virtuelles. Elle permet une efficacité accrue, une réduction des coûts, un déploiement plus rapide des charges de travail, une augmentation des performances d'application, une disponibilité de serveur en hausse, et l'élimination des complications liées à la gestion de serveurs ***(Bastien, 2019)***.

2.3.3 Avantages de la virtualisation

Les avantages de la virtualisation sont résumés comme suit :

- La portabilité

Une machine physique considère un serveur virtuel comme une suite de données. Un serveur virtuel peut donc être déplacé sans problème d'un matériel à un autre, sans prendre le risque de rencontrer des incompatibilités entre deux systèmes d'exploitation ou encore deux types de matériels. En virtuel, pas de risque de panne technique non plus.

Une fois les données considérées comme des fichiers, celles-ci peuvent être facilement transportées d'un espace de stockage à un autre. En cas de problème technique, il est alors possible de réinitialiser chaque serveur virtuel en l'installant sur un nouveau serveur physique.

- La disponibilité

La portabilité des données offre un autre avantage : celui de leur disponibilité. En créantun environnement virtuel à partir de plusieurs serveurs physiques, les données restent constamment disponibles. En cas de panne, il suffira alors de récupérer les serveurs virtuelsà partir d'un serveur physique fonctionnel. Le but est que ce type d'opération soit transparent pour les

utilisateurs, puisqu'une totale disponibilité des données est assurée.

- L'Efficacité

La virtualisation nous permet d'avoir une seule machine physique au service de plusieurs machines virtuelles. Cela signifie non seulement que nous avons besoin de moins de serveurs, mais aussi que nous pouvons utiliser ceux que nous avons au plus fort de leur capacité. Ces gains d'efficacité se traduisent par des économies de coût en matière de matériel, de refroidissement et de maintenance, sans oublier les avantages pour l'environnement d'une empreinte carbone réduite. La virtualisation nous permet également d'exécuter plusieurs types d'applications, de postes et de systèmes d'exploitation sur une seule machine, plutôt que de solliciter des serveurs distincts auprès de différents fournisseurs.

Nous ne dépendons ainsi plus de fournisseurs spécifiques, et la gestion de nos ressources physiques prend beaucoup moins de temps, ce qui permet à notre équipe informatique d'être plus productive.

- La flexibilité

Un logiciel de virtualisation offre à l'entreprise une plus grande souplesse dans la façon dont elle teste et attribue les ressources. Sauvegarder et restaurer des machines virtuelles est tellement simple que notre équipe IT peut tester et expérimenter facilement de nouvelles technologies.

La virtualisation nous permet également de créer une stratégie cloud en attribuant des ressources de machines virtuelles au sein d'un pool commun pour l'entreprise. Cette infrastructure cloud permet à notre équipe IT de contrôler qui accède à quelles ressources etsur quel appareil, améliorant ainsi la sécurité et la flexibilité.

- L'Agilité

Déplacement des systèmes d'exploitation de la même manière que nous pouvons déplacerun fichier ou une image d'un serveur physique vers un autre serveur physique.

- La fiabilité

La technologie de virtualisation nous permet de sauvegarder et de récupérer aisément nos données à l'aide de snapshots des serveurs existants. Nous pouvons également automatiser en

toute simplicité ce processus de sauvegarde afin de tenir toutes nos données à jour. En cas d'urgence, si nous avons besoin de restaurer nos données à partir d'une machine virtuelle sauvegardée, nous pouvons aisément migrer cette machine virtuelle vers un nouvel emplacement en quelques minutes.

Nous bénéficions ainsi d'une continuité d'activité et d'une reprise après sinistre, puisque nous pouvons facilement récupérer nos données en cas de perte.

- Tolérance aux pannes

Quand un serveur physique tombe en panne, le logiciel de gestion migre automatiquement les instances vers les serveurs disponibles si rapidement que nous ne réalisons même pas que le matériel physique est en panne ***(Alibaba, 2022)***.

2.3.4 Gestion des machines virtuelles

La virtualisation permet d'exécuter plusieurs systèmes d'exploitation sur un même ordinateur. Il faut d'abord installer un système d'exploitation spécial (l'hyperviseur) directement sur le matériel brut et installer les systèmes d'exploitation virtuels sur l'hyperviseur. Ces instances de ***'SE'*** s'appellent machines virtuelles ou ***'VM'***. Une seule machine physique peut en comprendre plusieurs dizaines, voire plusieurs centaines. Chaque ***'SE'*** se voit attribuer sa propre part de ressources physiques, isolées par un séparateur logique des autres ressources disponibles sur la machine invitée. Cette séparation des ressources est la tâche principale de l'hyperviseur, en plus de l'intégration des services de mise en cluster, de sauvegarde et d'autres ressources permettant l'existence d'hôtes multiples.

Les hyperviseurs les plus populaires actuellement sont fabriqués par Microsoft (Hyper-V), VMware (vSphere) et Citrix (XenServer).

Dans un serveur classique, il y a 3 couches : le hardware, l'OS et enfin la couche applicative. Avec la virtualisation, une couche supplémentaire est ajoutée au-dessus du hardware. Il s'agit de la couche de virtualisation réalisée par l'hyperviseur.

Il existe plusieurs types d'hyperviseur. Le plus répandu est celui présenté ci-dessus :

- ***Le type 1*** : dit « bare metal ». Il s'exécute au-dessus du hardware.
- ***Les hyperviseurs de type 2*** :« host métal » s'exécutent à l'intérieur d'un autre système d'exploitation ***(DELL, 2011).***

La figure **(2.2)** : se résume à faire fonctionner un ou plusieurs systèmes d'exploitation au lieu de pouvoir en installer qu'un par machine.

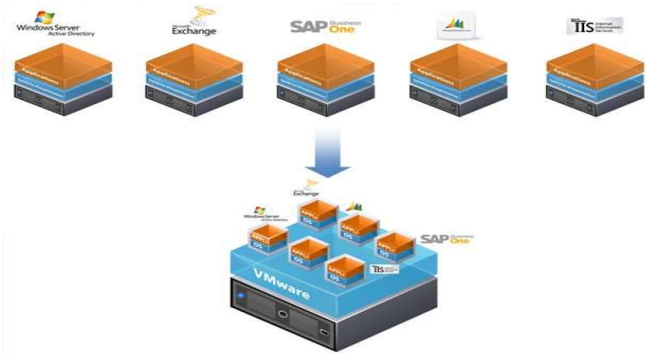


Figure 2.2:La virtualisation

2.3.5 Passage de la virtualisation au cloud computing

Une machine virtuelle, communément appelée VM, est un ordinateur défini par logiciel qui n'est pas différent d'un ordinateur physique. Le principal avantage des ordinateurs virtuels est qu'ils sont indépendants des ordinateurs physiques et hautement portables.

Ces caractéristiques rendent les machines virtuelles très populaires dans le monde d'aujourd'hui,même dans les environnements sur site et en particulier dans le cloud, d'où nécessite une migration des machines virtuelles vers le cloud computing.

Cette approche de la migration des machines virtuelles est la plus simple. Nous utilisons les technologies de virtualisation existantes telles que VMware pour configurer de nouvelles machines virtuelles en fonction de notre besoin.L'infrastructure existante est recréée dans le cloud et nous installons manuellementtous les logiciels nécessaires.

Dans ce type de migration, les utilisateurs ont la possibilité de réorganiser le système et d'exploiter davantage les avantages de la migration vers le cloud.

Tout d'abord, nous devons différencier trois types de migrations de machines virtuellesvers le cloud :

- Migration vers le cloud des machines virtuelles avec la configuration d'un nouvel environnement.
- Migration vers le cloud des machines virtuelles lift and shift avec une petite fenêtrede temps

d'arrêt.

- Soulevez et déplacez la migration vers le cloud des machines virtuelles avec unegrande fenêtre de temps d'arrêt.
- Selon le type, la quantité de travail et les outils varient ***(Odjel, 2021).***

2.4 Informatique en nuage (cloud computing)

Le passage de la virtualisation vers le cloud computing qui est une nouvelle technologie très populaire.

2.4.1 La définition de cloud computing

Le Cloud Computing (CC) définit un mode de structuration et externalisation des composants du système d'information de l'entreprise. Il est basé sur la technologie de virtualisation et automatisation, qui consiste à partager des ressources informatiques configurables d'un Data Center.

Ces ressources offrant des capacités de stockage de base de données, puissance de calcul,les réseaux, les serveurs d'applications, des logiciels de gestion de messagerie, et d'autres services sont mis à disposition par des sociétés tierces et accessibles, grâce à un système d'identification, via un PC et une connexion à Internet.

Le Cloud fournit aussi à ses clients des services à la demande, instantanés et élastiques avec facturation de ce qui a consommé, comme montrer la figure ci-dessus **(2.3)** ***(ZERTAL, 2020)***.

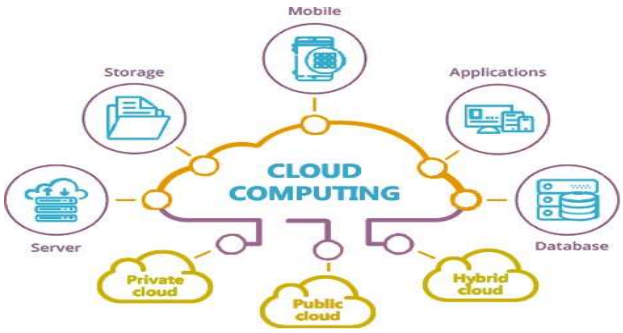


Figure 2.3:Cloud computing

2.4.2 Les avantages et les inconvénients du cloud

L'utilisation du cloud présente aussi bien des avantages que des inconvénients.

2.4.2.1 Les avantages du cloud computing

Les avantages du cloud computing sont ***(Astuces, 2022)*** :

- La flexibilité

Le cloud s'adapte à nos besoins au fur et à mesure qu'ils augmentent. Nous commandons l'espace de stockage ou la puissance de calcul dont nous avons besoin : inutile par exemple pour un particulier de payer un espace disque de 100 To alors que ses données ne représentent que quelques Go. Une fois que nous avons atteint notre quota d'espace disque, il est très simple de passer à la formule supérieure pour avoir plus d'espace de stockage. Nous payons uniquement ce dont nous avons besoin et pouvons facilement ajuster notre abonnement selon nos besoins.

- L'accessibilité

Il nous offre grâce à un support et une connexion Internet une totale accessibilité à nos applications et services de stockage. Il suffit d'une connexion internet pour accéder à ses données ou logiciels, le cloud est ainsi particulièrement adapté au milieu professionnel et au télétravail.

- L'absence de financement initial

Nous n'avons pas besoin de faire un lourd investissement, qu'il soit matériel ou applicatif, pour mettre en place ce type de solution de stockage. Ceci reste la responsabilité du prestataire. Nous ne payons qu'un abonnement mensuel. C'est une manière utile de mieux gérer notre budget, car nous ne payons que ce que nous consommons pendant cette période.

2.4.2.2 Les inconvénients du cloud computing

Les inconvénients du cloud computing sont ***(Astuces, 2022)*** :

- La fiabilité et la localisation du cloud

Lorsque le cloud n'est pas hébergé dans les meilleures conditions, la sécurité de nos données n'est pas garantie. Il nous faudra donc bien nous renseigner sur les mesures de chiffrement de nos données ainsi que sur les mesures de sécurité garanties par notre prestataire cloud.

- La nécessité d'avoir une connexion internet

Pour accéder au cloud, nous devons disposer d'une connexion internet pour accéder à nos données et applications. Si ce n'est pas le cas, nous ne serons pas en mesure d'accéder à nos données**.**

2.4.3 Les services du Cloud Computing

Les fortes avancées dans le domaine de la virtualisation ont rendu possible le Cloud Computing. Cette virtualisation permet d'optimiser les ressources matérielles en les partageants entre plusieurs environnements « time-sharing ». De même, il est possible de coupler l'application avec son système d'exploitation et le matériel encapsulé dans lamachine virtuelle. Cela assure également un « provisionning », c'est-à-dire la capacité de déploiement d'environnement, de manière automatique.

Le Cloud Computing couplé, aux technologies de virtualisation, permet la mise à disposition d'infrastructures et de plate-forme à la demande. Mais le Cloud Computing ne concerne pas seulement l'infrastructure (IaaS), il bouleverse la plate-forme d'exécution (PaaS) et les applications (SaaS) : Comme nous le verrons plus loin, le Cloud est à la fois transversal et vertical.

- Iaas (Infrastructure as a Service)

Il s'agit de la mise à disposition, à la demande, de ressources d'infrastructures dontla plus grande partie est localisée à distance dans des Datacenters. L'IaaS permet l'accès auxserveurs et à leurs configurations pour les administrateurs de l'entreprise. Le client a la possibilité de louer des clusters, de la mémoire ou du stockage de données. Le coût est directement lié au taux d'occupation. Une analogie peut être faîte avec le mode d'utilisation des industries des commodités (électricité, eau, gaz) ou des Télécommunications.

- PaaS (Platform as a Service)

Il s'agit des plateformes du nuage, regroupant principalement les serveurs mutualisés etleurs systèmes d'exploitation. En plus de pouvoir délivrer des logiciels en mode SaaS, le PaaS dispose d'environnements spécialisés au développement comprenant les langages, les outils et les modules nécessaires.

L'avantage est que ces environnements sont hébergés par un prestataire basé à l'extérieurde

l'entreprise ce qui permet de ne disposer d'aucune infrastructure et de personnel de maintenance et donc de pouvoir se consacrer au développement.

- Saas (Software as a Service)

Concept consistant à proposer un abonnement à un logiciel plutôt que l'achat d'une licence. On oublie donc le modèle client-serveur et aucune application n'est installée sur l'ordinateur, elles sont directement utilisables via le navigateur Web.

L'utilisation reste transparente pour les utilisateurs, qui ne se soucient ni de la plateforme, ni du matériel, qui sont mutualisés avec d'autres entreprises. Le Saas remplace l'ASP (application service provider), aussi appelé fournisseur d'applications hébergées ou FAH, qui est une entreprise qui fournit des logiciels ou des services informatiques à ses clients au travers d'un réseau.

Deux principales différences avec l'ASP traditionnel sont qu'une simple interface web est utilisée côté client dans tous les cas (pas de client lourd), et que le SaaS propose une seule instance de logiciel qui évolue indépendamment des clients ***(Lucas, 2010).***

La figure **(2.4)** résumé tous les services du cloud computing.

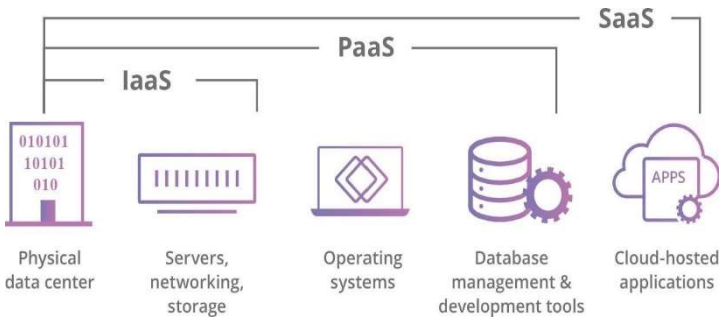


Figure 2.4: Les services du cloud computing

Sur le tableau **(2.1),** on résume les avantages et les inconvénients des services ***(Lucas, 2010).***

Tableau 2. 1:Avantages et inconvénients des services

	Avantage	Inconvénient
IaaS	• Grande flexibilité. • Administration. • Personnalisation.	• Sécurité. • Besoin d'un administrateur systeme
PaaS	• Le déploiement est automatisé. • Pas de logiciel supplémentaireà acheter ou à installer. • Environnement hétérogène. Pas d'infrastructure nécessaire.	• Pas de contrôle desmachines virtuellesSous-jacentes. • Convient uniquement aux applications Web. Limitation des langages.
SaaS	• Plus d'installation. • Plus de mise à jour. • Plus de migration de données. • Paiement à l'usage. Test de nouveaux logiciels avec facilité.	• Réactivité des applicationsWeb pas toujours idéals. • Pas de contrôle sur le stockage. • Pas de contrôle sur lasécurisation des données associées au logiciel. Dépendance des prestataires.

2.4.4 Types de cloud computing

Il existe trois modes de déploiement de services cloud : le cloud public, le cloud privé etle cloud hybride.

- Cloud public

Un cloud public est détenu et exploité par un fournisseur de services cloud tiers, qui propose des ressources de calcul, telles que des serveurs et du stockage, via Internet. Microsoft Azure est un exemple de cloud public. Dans un cloud public, tout le matériel, tousles logiciels et toute l'infrastructure sont la propriété du fournisseur du cloud. Nous accédons àces services et nous gérons notre compte par l'intermédiaire d'un navigateur web.

- Cloud privé

Le cloud privé est l'ensemble des ressources de cloud computing utilisées de façon exclusive par une entreprise ou une organisation. Le cloud privé peut se trouver physiquement dans le centre de données local de l'entreprise. Certaines entreprises paient également des fournisseurs de services pour qu'ils hébergent leur cloud privé.

Le cloud privéest un cloud dans lequel les services et l'infrastructure se trouvent sur un réseau

privé.

- Cloud hybride

Le cloud hybride regroupe des clouds publics et privés, liés par une technologie leur permettant de partager des données et des applications. En permettant que les données et applications se déplacent entre des clouds privé et public. Un cloud hybride offre à notre entreprise une plus grande flexibilité, davantage d'options de déploiement et une optimisation de notre infrastructure, de notre sécurité et de notre conformité existante ***(Microsoft, 2022).***

2.4.5 Le rôle de cloud computing

La transformation est une tendance constante, qui devient un besoin absolu de l'heure dans le monde trépidant d'aujourd'hui. Avec la technologie qui transforme chaque bit d'information dans un nouveau format raffiné, il y a beaucoup de possibilités en matière destockage et de manipulation des données.

Alors que les téléphones intelligents et les médias sociaux commencent à dominer le perchoir, il y a beaucoup de conversations autour de ce qui va suivre. La réponse évidente de l'heure est l'Internet des objets ou IoT. Avec Internet produisant d'énormes volumes de données chaque seconde, l'infrastructure de données est sous pression, ce qui rend nécessairela recherche de solutions pour faciliter l'utilisation du stockage de données.

Depuis l'essor du Cloud, il y a une évolution massive vers son utilisation comme moyende stockage pour les particuliers et les entreprises. Compte tenu de l'évolutivité et de la dynamique des données, on insiste beaucoup sur l'utilisation du Cloud computing pour rendre les données disponibles à distance.

Mettant à profit cette évolutivité, le Cloud s'est avéré être un outil efficace pour transférerdes données via les canaux Internet traditionnels ainsi que via une liaison directe dédiée. Laméthode traditionnelle n'est pas largement préférée ; cependant, dans le même temps, de nombreuses entreprises préfèrent utiliser le lien direct pour transférer les données vers le Cloud, compte tenu de la qualité des données et de la sécurité qu'elle assure lors de la phasede transfert.

Ce n'est pas tout ; le Cloud est devenu partie intégrante du monde Internet. En termes simples, le cloud peut être qualifié de catalyseur en matière d'IoT. Le Cloud est sans aucun doute une solution idéale pour répondre à tous les besoins data-driven des entreprises. Au fur et à mesure que cette technologie se développe, elle fournit une plate-forme agile aux

développeurs pour créer des applications significatives afin d'établir de meilleurs dispositifsde données sur Internet ***(Id Excel, 2017)*.**

2.4.6 Le fonctionnement de cloud computing

Pour comprendre ce qu'est le cloud dans sa partie plus technique, il faut le segmenter en plusieurs éléments qui le composent.

- Le matériel

Cette technologie repose en grande partie sur le socle de la machine virtuelle (appelée «VM » ou encore « Serveur Virtuel Privé »), mais nécessite tout de même une infrastructurephysique. C'est ainsi que naissent les datacenters : des locaux de grande capacité allant jusqu'à représenter la surface d'une dizaine de terrains de foot, composés de serveurs.

- La plateforme

➢ *Un serveur cloud est segmenté en trois grandes catégories disposées en forme de couche :*

✓ ***Le CPU ou la RAM***

Représente la partie du serveur utile à la réalisation de calculset incarnant la zone permettant d'effectuer le traitement des opérations.

✓ ***Le Network***

Permet de gérer les différents échanges avec les réseaux Internet.

✓ ***Le stockage***

Représente la partie du serveur qui offre la possibilité de faire du stockage brut, c'est-à-dire du stockage sur du moyen ou long terme. Pour illustrer cepoint, il faut prendre l'exemple de la boîte mails, qui stocke un grand nombre de mailsouvent garder sur une longue période. Chacune de ses couches possèdent une utilitépropre, et/ou peut être combinée.

- Les VM ou VPS (Serveur Virtuel Privé)

C'est là qu'apparaissent les VM et VPS. Souvent perçus comme des termes barbares, les VM et VPS sont similaires mais ne sont pas utilisés par la même cible. Les hébergeurs et professionnels informatiques parleront de VM, signifiant « Virtual machine » ; alors que ce seront les clients qui parleront de VPS, qui lui veut dire « Virtual Private Server », terme utilisé

lorsqu'ils consomment cette solution.

2.4.7 La flexibilité et l'adaptation à la demande

Plutôt que d'exploiter un service physique dans ses locaux, le cloud permet de virtualiseret de s'adapter aux demandes de son utilisateur. Il est souvent qualifié de flexible et d'adaptable. Le cloud offre une pluralité de possibilités à ses utilisateurs, qui pourront ainsistocker un nombre volumineux de données en exploitant la puissance de calcul et de stockage de celui-ci ***(Emilie, 2016).***

2.4.8 Les caractéristiques de cloud computing

Le modèle Cloud Computing se différencie par les cinq caractéristiques essentielles suivantes :

- Accès aux services par l'utilisateur à la demande

La mise en œuvre des systèmes est entièrement automatisée et c'est l'utilisateur, au moyen d'une console de commande, qui met en place et gère la configuration à distance.

- Accès réseau large bande

Ces centres de traitement sont généralement raccordés directement sur le backbone Internet pour bénéficier d'une excellente connectivité. Les grands fournisseurs répartissent les centres de traitement sur la planète pour fournir un accès aux systèmes en moins de 50 ms de n'importe quel endroit.

- Réservoir de ressources (non localisées)

La plupart de ces centres comportent des dizaines de milliers de serveurs et de moyens de stockage pour permettre des montées en charge rapides. Il est souvent possible de choisirune zone géographique pour mettre les données "près" des utilisateurs.

- Redimensionnement rapide (élasticité)

La mise en ligne d'une nouvelle instance d'un serveur est réalisée en quelques minutes, l'arrêt et le redémarrage en quelques secondes. Toutes ces opérations peuvent s'effectuer automatiquement par des scripts. Ces mécanismes de gestion permettent de bénéficier pleinement de la facturation à l'usage en adaptant la puissance de calcul au trafic instantané.

- Facturation à l'usage

Il n'y a généralement pas de coût de mise en service (c'est l'utilisateur qui réalise les

opérations). La facturation est calculée en fonction de la durée et de la quantité de ressources utilisées. Une unité de taitement stoppée n'est pas facturée ***(Vahatrainiravo,2015)*.**

2.4.9 La comparaison entre la virtualisation et le cloud computing

Étant donné une compréhension générale du fonctionnement de la virtualisation et du cloud computing, voyons en quoi ils diffèrent l'un de l'autre. Voici une comparaison directeentre les deux basée sur sept domaines clés.

Notez que dans ces comparaisons, nous excluons les clouds privés car, dans de tels environnements, les organisations traitent directement avec les hyperviseurs et le matériel physique sous-jacent comme elles le feraient avec un environnement de virtualisation. Au contraire, dans les clouds publics (SaaS, IaaS, PaaS), c'est le fournisseur de cloud qui gère ces éléments.

Sur le tableau **(2.2),** on a présent la comparaison des deux reseaux traditionnelle qui sont la virtualisation et cloud computing (nuage) ***(Carlin, 2021) (Redhat,2018).***

Tableau 2. 2:Comparaison entre la virtualisation et le cloud computing

Secteur clé	Virtualisation	Cloud computing
Définition	Technologie	Méthodologie
Objet	Créer plusieurs environnements simulés àpartir d'un même système physique	Regrouper et automatiser des ressourcesvirtuelles pour une utilisation à la Demande
Type de service	Licence	SaaS, IaaS, PaaS
Utilisation	Fournir des ressources en paquets à desutilisateurs spécifiques pour une tâche spécifique	Fournir des ressources variables à desgroupes d'utilisateurs pour diverses tâches
Installation	Installé sur des serveurs sur site. Avecl'infrastructure à portée de main, la configuration nécessite généralement l'installation d'un logiciel hyperviseur pour créer et gérer les machines virtuelles, ainsi que les systèmes d'exploitation et les applications qui s'exécutent dessus.	Aucune installation nécessaire pour lesplates-formes SaaS, alors que les services IaaS et PaaS nécessitent l'installation et la configuration de logiciels (par exemple, les systèmes d'exploitation et les applications), maispas le logiciel hyperviseur.
Configuration	À partir d'une image.	À partir d'un modèle.
Durée de vie	Années (long terme).	Heures ou mois (court terme).
Évolutivité	Peut prendre en charge les machines virtuelles en fonction de la capacité dumatériel physique. D'autres machines hôtes peuvent être ajoutées, mais cela nécessiterait des dépenses supplémentaires pour l'infrastructure Physique. Évolutivité verticale.	La capacité du cloud est pratiquement illimitée, le fournisseur de services prenant en charge les exigences matérielles. Permet aux entreprises d'augmenter ou de réduire leur taille enfonction de leurs besoins. Évolutivité horizontale.
Charge de travail	Avec état	Sans état
Souplesse	Bien que nous puissions facilement démarrer ou arrêter des machines virtuelles, nous disposons toujours d'uneflexibilité limitée si nous considérons lesexigences matérielles substantielles.	Flexibilité optimale, car nous n'avons pas ànous soucier du matériel. Différents types de déploiements cloud et de modèles de livraison permettent aux entreprises d'essayer différentes solutions et de les mettre à niveau.
Type d'architecture	Client unique.	Multi-client.
Matériel Exigence	Nécessite du matériel dédié pour créer plusieurs machines virtuelles.	Aucun matériel nécessaire.
L'intégration	Peut s'intégrer aux clouds privés et publics, aux bases de données, aux appareils IoT et aux applications héritées(avec des logiciels supplémentaires).	Peut s'intégrer aux solutions et applications existantes, mais les logiciels hérités peuvent être délicats etnécessiter une solution de contournement.
Coût	Dépenses d'investissement élevées,dépenses d'exploitation faibles.	***Cloud privé :*** dépenses d'investissement élevées, dépenses d'exploitation faibles. ***Cloud public*** : dépenses d'investissement faibles, dépenses d'exploitation élevées.
Reprise après sinistre	Permet aux entreprises de créer plusieurs redondances d'une machine virtuelle pouréviter les temps d'arrêt prolongés même en cas de panne d'un serveur.	La plupart des fournisseurs de servicescloud gèrent plusieurs centres de données à travers le monde pour s'assurer que les organisations/utilisateurs ont accès à leurs données à tout moment.

La virtualisation nécessite des serveurs physiques pour fournir les ressources aux machines virtuelles. Par conséquent, l'augmentation du nombre de machines virtuelles au- delà de la capacité du matériel existant signifie que l'entreprise devra acquérir des serveurs supplémentaires pour les prendre en charge. Avec une solution cloud, cependant, ajouter plus de machines virtuelles signifie simplement louer plus de ressources auprès de notre fournisseur de services cloud, ce que nous pouvons faire sans trop d'effort en quelques minutes. En termes de coûts et d'évolutivité, le cloud est l'option la plus simple et la plus rentable, car il ne nécessite pas beaucoup de temps ou d'investissement supplémentaire pourévoluer.

Une organisation peut choisir d'utiliser la virtualisation pour créer un cloud privé interne(un cloud créé au sein de l'organisation à l'aide de son propre matériel). Cependant, il s'agitd'une entreprise plus coûteuse par rapport à un cloud privé hébergé, qui est utilisé uniquement par une entreprise mais hébergé par un tiers, ou un cloud public, qui est partagéavec d'autres entreprises. La raison principale pour laquelle nous ferions cela est si, peut-êtreen raison de problèmes de confidentialité, nous souhaitons avoir plus de contrôle sur nos machines virtuelles et nos données ***(Carlin, 2021).***

La figure (2.5) donne une simple comparaison entre la virtualisation et cloud compu

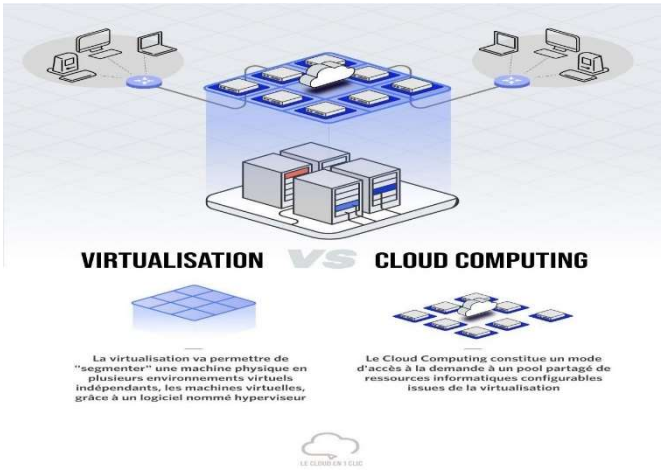


Figure 2.5:Comparaison entre la virtualisation et cloud computing

2.4.10 Un besoin de réseau programmable

L'expérience des réseaux de télécommunication montre qu'il n'existe pas de protocole unique dont les caractéristiques sont telles qu'ilest idéal dans toutes les situations (pour tous supports et pour toutes les applications). Par exemple, TCP tel que nous le connaissons actuellement sur IP, n'est pas idéal dans des environnements sans fil et satellitaires et IP n'est pas adapté pour

des conditions de mobilitéforte et des procédures handoff. Partant de ce constat, il est nécessaire de tenter au niveau des architectures non pas de forcer des protocoles donnés, mais de rendre celle-ci ouvertesafin de permettre le déploiement dynamique de protocoles existants adaptés et le cas échéant de nouvelles solutions.

L'objectif principal des réseaux programmables et d'aboutir à la définition d'un système d'exploitation réseau offrant des interfaces de programmation ouvertes permettant la conception, le développement et l'introduction rapide de nouveaux services. Concrètement,ce type d'architecture doit permettre aux opérateurs et fournisseurs de services de répondre rapidement au besoin des usagers et d'innover constamment dans le domaine des services offerts ceci bien sûr à moindre coût.

Les réseaux programmables ont, ces dernières années, retenus l'attention de nombreux chercheurs et industriels de télécommunication. De nombreux projets ont aboutis à des résultats d'une qualité irréprochable permettant à ce type d'architecture d'entamer une phase d'industrialisation, aujourd'hui bien avancée. Cette section est consacrée à cesprojets. Cette dernière détaille les principales architectures étudiées à travers le monde et présente les différents efforts de standardisation en cours.

2.5 Conclusion

Avec la croissance constante des infrastructures de réseau actuelles, la virtualisation et le cloud computing ne résolvent pas toutes les problématiques car les réseaux numériques deviennent de plus en plus grands et complexes. Et le degré de virtualisation, cloudcomputing, le désir de flexibilité et de l'évolutivité maximale s'accroissent également. Ces derniers n'ont pas été en mesure de répondre à des exigences de réseau et même de sécuriser les données vulnérables aux attaques. Pendant un certain temps, ce qui a conduit au développement du concept de réseau défini par logiciel appelé SDN.

Dans le chapitre suivant, nous traitons la technologie SDN et ses composants dans les réseaux.

Chapitre 3

Les réseaux Programmables SDN

Chapitre 3 Les réseaux programmables SDN

3 Introduction

Depuis une dizaine d'années, les réseaux SDN sont de plus en plus utilisés en particulierdans les Datacenters des fournisseurs de Cloud, mais aussi par les opérateurs sur leur WAN et au sein des infrastructures réseaux des grandes entreprises. Google par exemple a été à lapointe en connectant ses *Data centers* avec des commutateurs et des contrôleurs mettant en œuvre le protocole *Openflow*. D'autres groupes ont aussi annoncé l'utilisation d'Openflow comme Amazon, Microsoft et AT&T.

Les réseaux SDN facilitent la gestion des réseaux en environnements Cloud en permettant automatisation, évolution rapide, redondance et segmentation virtuelle des clients en optimisant les coûts d'infrastructure ***(prosica).***

Dans ce chapitre, nous allons faire une présentation simple du SDN et Openflow, afin de mieux faire comprendre à toute personne cette évolution actuelle des réseaux qui leur permettra de se faire leur propre opinion en toute connaissance de cause.

3.1 Définition et architecture SDN

Dans cette section on va donner la définition et l'architecture SDN.

3.1.1 Définition SDN

Le SDN est un nouveau paradigme qui décrit une architecture réseau dont le plan de contrôle est totalement découplé du plan de données.

Selon l'*ONF* (Open Network Fondation), SDN est une architecture qui sépare le plan de contrôle du plan de données, et centralise toute l'intelligence de réseau dans une entité programmable appelé « Contrôleur », afin de gérer plusieurs éléments du plan de données (Ex switches ou routeurs, etc.) via des *APIs* (Application Programming Interface).

Plus concrètement, on peut dire qu'une architecture réseau suit le paradigme SDN si, et seulement si, elle vérifie les points suivants ***(PARKOO, 2015)*** :

- Le plan de contrôle est complètement découplé du plan de données, cette séparation est matérialisée à travers la définition d'une interface de programmation (Southbound API).
- Toute l'intelligence du réseau est externalisée dans un point logiquement centralisé appelé contrôleur SDN. Ce dernier offre une vue globale sur toute l'infrastructure physique.
- Le contrôleur SDN est un composant programmable qui expose une API (NorthboundAPI)pour spécifier des applications de contrôle*)*.

La figure (3.1) montre la différence et la migration d'un réseau traditionnel vers le réseau SDN.

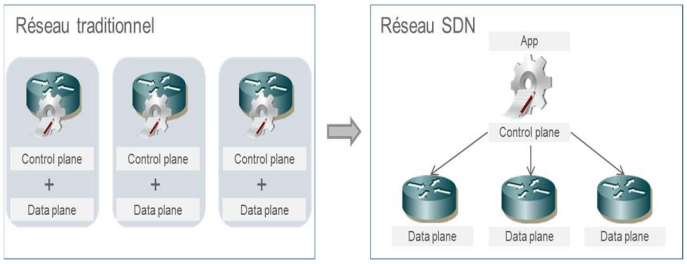


Figure 3.1:Réseau traditionnel vers le réseau SDN

3.1.2 Architecteur SDN

La structure de SDN se compose de trois parties principales. Le niveau le plus bas, inclutle Data plane, au plus haut niveau dont l'existence de l'application plane, et le control plane se trouve entre les deux, comme le montre la figure (3.2).

La communication entre les contrôleurs et le Data plane est gérée via le SBI (South-Bound Interface), qui se trouve au niveau de commutateur SDN. Et la communication entre les applications et le contrôleur est assurée par NBI (Nourth-Bound Interface), qui se trouvedans le control plane.

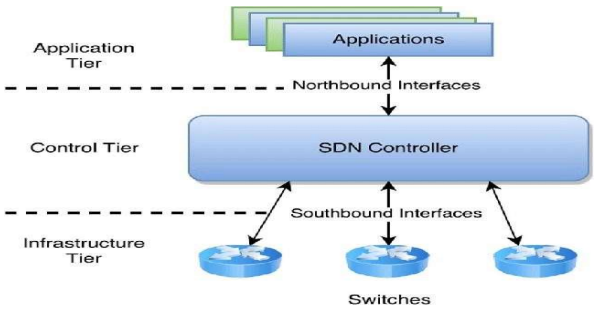


Figure 3.2:Architecteur SDN

3.1.2.1 La couche application

La couche application concerne tout ce qui touche l'administration de l'équipement, il s'agit donc des applications SDN conçues pour répondre aux exigences d'une infrastructure,À travers la plate-forme programmable fournie par la couche contrôle, les applications SDNpeuvent accéder et contrôler les dispositifs de commutation dans la couche infrastructure, etd'introduire de nouvelles fonctions et de services.

Exemple d'applications SDN :

- Application d'équilibrage de charge.
- Virtualisation du réseau.

3.1.2.2 La couche contrôle

La couche contrôle est logicielle, basée sur le contrôleur SDN, permet de contrôler le plan de données (la couche infrastructure) en établissant des règles qu'il devra suivre. La couche contrôle relie l'infrastructure du réseau avec la couche application à travers ses deux interfaces.

L'interaction avec la couche physique se fait à travers une interface **sud** (The South- Bound interface) qui permet de :

- Importer les règles de transmission de paquets à partir du contrôleur pour désigner le comportement des équipements physiques de réseau.
- Envoyer l'état du réseau au contrôleur.
- Supporter le protocole OpenFlow qui se positionne comme une API sud agissant directement sur le plan de données.
- Etablissement des tables de routages.

L'interaction avec la couche application se fait à travers une interface vers le **nord** (The North-Bound interface). Elle fournit des points d'accès aux services sous diverses formes, par exemple : une interface de programmation d'applications (API). Dans le cas du SDN, ces applications ont accès aux informations reportées par le switch et peuvent ainsi prendre des décisions en définissant les règles de transfert de paquets aux switches, et tout cela grâce aux API.

3.1.2.3 La couche infrastructure

Cette couche est principalement constituée des équipements physiques ou, et virtuels du réseauqui sont responsables de l'acheminement du trafic, tel que les switches et les routeurs qui sont interconnectés pour former un seul réseau ***(LAMARA, 2018).***

La fonction principale de ses équipements :

- La collecte des états du réseau (la topologie actuelle du réseau, statistiquesdu trafic sur le réseau, la densité d'utilisation du réseau) et les envoyer aucontrôleur.
- Le traitement des paquets se basant sur les règles de gestion fournit par lecontrôleur.
- Les équipements physiques sont interconnectés à travers différents supports de transmission, comme les files de cuivre, fibre optique, les faisceaux hertzien.
- Acheminement, fragmentation et réassemblage des paquets ***(OULD LAMARA, 2018)***.

➢ **Interfaces de communications**

Il existe principalement trois types d'interfaces permettent aux contrôleurs de communiquer avec leur environnement : interface **Sud**, **Nord** et **Est/Ouest**.

o **Interfaces Sud**

Les interfaces Sud ou (South-Bound APIs) représentent les interfaces de communication,qui permettent au contrôleur SDN d'interagir avec les équipements de la couche d'infrastructure, tel que les switches, et les routeurs.

Le protocole le plus utilisé, et le plus déployé comme interface Sud est le protocole OpenFlow, qui a été standardisé par l'ONF. Sa dernière version est 1.5, plus de détails sur ce protocole sera donnée dans la prochaine section. Il existe dorénavant d'autres alternatives d'interface Sud, tels que *Forces*, ou *Open Vswitch, Data base* (*OVSDB*), mais le protocole *openflow* est actuellement le standard de facto, qui est largement accepté et répandu dans les réseaux SDN.

o **Interfaces Nord**

Les interfaces Nord servent à programmer les équipements de transmission, en exploitant l'abstraction du réseau fourni par le plan de contrôle. Il est noté que contrairement à la South-Bound API qui a été standardisé, l'interface nord reste encore une question ouverte. Bien que la nécessité d'une telle interface standardisée constitue un débat considérable au sein de

l'industrie, une API nord ouverte permette plus d'innovation et d'expérimentation.

Plusieurs implémentations de cette interface existent, chacune de ces implémentations offre des fonctionnalités bien différentes. Le RESTful considéré comme l'API nord le plus répandue dans les réseaux SDN.

- **Interfaces Est/Ouest**

Les interfaces Est/Ouest sont des interfaces de communication qui permettent la communication entre les contrôleurs dans une architecture multi-contrôleurs, pour synchroniser l'état du réseau. Ces architectures sont très récentes et aucun standard decommunication inter-contrôleur n'est actuellement disponible ***(Choukri, 2019)***.

3.2 Avantages de SDN

Dans cette section on va résulter les avantages de SDN :

3.2.1 Réseaux programmables

Avec le SDN, il est plus simple et moins sujet aux erreurs de modifier les politiques réseaux, puisque nous devons juste changer une politique de haut niveau et non plusieurs règles.

De plus, la réaction à une modification du réseau (une nouvelle interconnexion ou un nouveau périphérique par exemple) ou un événement réseau (comme une attaque ou une suspicion d'intrusion) est simple et efficace, puisque nous pouvons facilement modifier l'ensemble du réseau pour prendre ce changement en ordre. De plus, cela peut se faire au plus près de la source et éviter une charge inutile dans notre réseau interne.

Avant le SDN, nous devions modifier manuellement les règles qui peuvent entraîner des erreurs ou ralentir les temps de réaction.

Enfin, la centralisation de la logique dans un tel contrôleur entièrement personnalisé avecdes connaissances globales et une grande puissance de calcul simplifient le développementde plus fonctions sophistiquées.

Cette capacité de programmation du réseau est l'élément clé du SDN.

3.2.2 Routage

SDN peut également être utilisé pour gérer les informations de routage de manière centralisée, en déléguant le routage et en utilisant une interface pour le contrôleur.

3.2.3 Politique unifiée

Avec son contrôleur, SDN garantit également un réseau unifié : puisque le contrôleur est responsable de l'ajout de règles calculées dans les commutateurs, il n'y a aucun risque qu'un opérateur réseau oublie un switch ou installe des incohérences règles entre les appareils.

En effet, l'opérateur va juste spécifier une nouvelle règle et le contrôleur adaptera la configuration pour envoyer des règles cohérentes dans chaque appareil concerné.

De plus, si un appareil est déplacé, les anciennes règles peuvent être supprimées et les nouvelles sont calculées et ajoutées dynamiquement. Ainsi, il est facile de déployer une politique unifiée mais également pour ajouter une automatisation intelligente dans le réseau.

3.2.4 Flexibilité

SDN apporte également une grande flexibilité dans la gestion du réseau. Il devient facile de rediriger le trafic, d'inspecter des flux particuliers, de tester de nouvelles stratégies ou de découvrir des flux inattendus.

Donc il permet d'augmenter le taux d'innovation au niveau de l'infrastructure réseau, et cela conduit à l'apparition de nouvelles idées. Par exemple, lesdéveloppeurs peuvent tester des applications au sein du réseau sans affecter les performancesdu réseau ou des services. Ce qui nous permet d'obtenir des performances plus élevées par rapport à celles qui existent actuellement ***(PARADIS, 2014)***.

3.2.5 Gestion du cloud

L'utilisation d'une approche SDN hybride et de conception cloud peut fournir, à un responsable informatique averti, l'agilité économique nécessaire pour répondre aux besoins d'infrastructure de l'entreprise tout en lui permettant, de résoudre de manière proactive, les principaux problèmes de sécurité ***(Strauss, 2017)***.

3.2.6 Simplification matérielle

Une nouvelle façon d'optimiser les périphériques matériels est déployée lors de l'utilisationdu SDN. Tout le matériel existant et nouveau peut être affecté à un objectif spécifique

grâce à l'utilisation de contrôleurs SDN. Par conséquent, il élimine la restriction des périphériquesmatériels là où il est dédié à un seul objectif**.**

3.3 Protocole openflow dans architecteur SDN

OpenFlow est un protocole qui permet à un serveur d'indiquer aux commutateurs réseauoù envoyer les paquets. Dans un réseau conventionnel, chaque commutateur possède un logiciel propriétaire qui lui dit quoi faire. Avec OpenFlow, les décisions de déplacement depaquets sont centralisées, de sorte que le réseau peut être programmé indépendamment descommutateurs individuels et de l'équipement du centre de données.

Dans un commutateur conventionnel, le transfert de paquets (le chemin des données) et le routage de haut niveau (le chemin de contrôle) se produisent sur le même appareil. Un commutateur OpenFlow sépare le chemin de données du chemin de contrôle. La partie du chemin de données réside sur le commutateur lui-même ; un contrôleur séparé prend des décisions de routage de haut niveau. Le commutateur et le contrôleur communiquent au moyen du protocole OpenFlow. Cette méthodologie, connue sous le nom de réseau défini par logiciel (SDN), permet une utilisation plus efficace des ressources du réseau que ce quiest possible avec les réseaux traditionnels. OpenFlow a gagné en popularité dans des applications telles que la mobilité VM (machine virtuelle), les réseaux critiques et les réseaux mobiles IP de nouvelle génération.

Il est standardisé par l'Open Networking Foundation (ONF) et implémenté par de nombreux équipementiers, dont HP, Cisco (ACI format propriétaire), IBM, Juniper, NEC etEricsson ***(Studios, 2013)*.**

3.4 La genèse d'openflow

L'histoire d'OpenFlow est intéressante et permet de mieux comprendre son rôle fondamental dans la conception de l'architecture SDN (Software Defined Network) et la virtualisation des fonctions réseau.

OpenFlow a été initié comme un projet à l'université de Stanford, lorsqu'un groupe de chercheurs exploraient la manière de tester de nouveaux protocoles dans le monde IP (créerun réseau expérimental confondu avec le réseau de production), mais sans arrêter le trafic du réseau de production lors des tests.

C'est dans cet environnement que les chercheurs à Stanford ont trouvé un moyen de séparer

le trafic de recherche du trafic du réseau de production qui utilise le même réseauIP.

Ils ont découvert que bien que les constructeurs de matériel réseau conçussent leurs produits différemment, tous utilisaient des tables de flux (flow table) afin d'implanter les services réseau tels que les NATs, la QoS, les firewalls, les systèmes de deep packet inspection, etc. Par ailleurs, bien que l'implantation des tables de flux différent entre ces constructeurs, les chercheurs ont découvert qu'ils pouvaient exploiter un ensemble de fonctions communes.

Le résultat de l'équipe de recherche à Stanford a été OpenFlow, qui fournit un protocole ouvert (open Protocol) qui permet aux administrateurs de réseau de programmer les tables de flux (flow tables) dans leurs différents routeurs IP et commutateurs Ethernet, dans un but(dans le cas de Stanford) de séparer le trafic de recherche du trafic de production, chacun avec son ensemble de fonctionnalités et caractéristiques de flux ***(EFORT, 2016)***.

3.5 Structure d'un commutateur openflow

OpenFlow propose une nouvelle architecture bien adaptée aux environnements des réseaux virtuels, l'idée principale est de faire communiquer les deux plans qui ont été séparés, le contrôleur quant à lui peut desservir l'ensemble du réseau par le biais d'OpenFlow en utilisant les liaisons du réseau.

La figure (3.3) détaille les éléments d'un réseau OpenFlow.

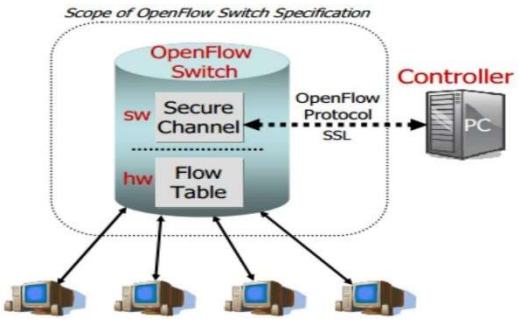


Figure 3.3:Architecture d'un switch openflow

Le réseau est formé essentiellement d'un ou plusieurs commutateurs OpenFlow, un ou plusieurs contrôleurs, et finalement le protocole OpenFlow définissant tous les messages échangés entre les contrôleurs et les commutateurs et permettant de standardiser la communication.

3.5.1 Contrôleur openflow

Un contrôleur SDN est le point stratégique (cerveau du réseau) d'un réseau défini par

logiciel. Un contrôleur open flow est un type de contrôleur SDN qui utilise le protocole openflow pour connecter et configurer les dispositifs (routeurs, commutateurs, etc.). Afin de déterminer le meilleur chemin pour le Trafic des applications.

3.5.2 Switch OpenFlow

Les éléments de base du switch OpenFlow sont les tables OpenFlow. Ces tables vont assurer la classification des paquets en se basant sur plusieurs champs des entêtes niveau 2,3ou 4.

On distingue deux types de switch OpenFlow :

- ***Switch OpenFlow-only :*** Supporte uniquement les actions requises pour lefonctionnement du protocole OpenFlow.
- ***Switch OpenFlow-enabled :*** En plus des actions requises pour le protocoleOpenFlow, ce switch supporte les actions d'un switch ordinaire.

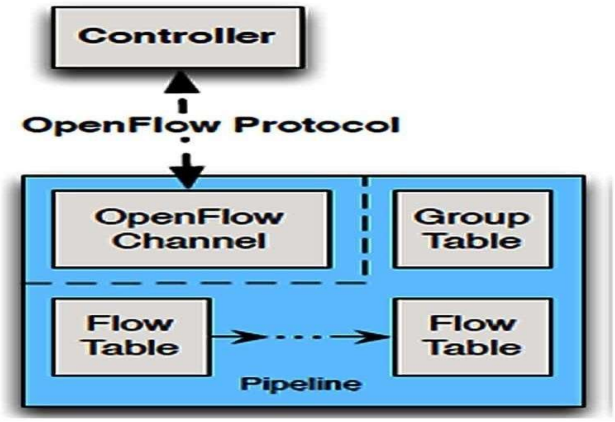


Figure 3.4:Elements du switch openflow

La figure (3.4) illustre les principales fonctions d'un commutateur OpenFlow.

3.5.3 Le canal sécurisé

Le canal sécurisé est l'interface qui connecte chaque switch OpenFlow à un contrôleur. Cette interface permet au contrôleur de recevoir les messages du switch et de pouvoir la gérer à travers le réseau. Le canal doit être sécurisé afin d'assurer le bon déroulement des communications entre le switch et le contrôleur. Pour cela l'échange de messages se fait au cours d'une session TCP établie via le port 6653 du serveur contrôleur ou à travers une connexion SSL/TLS (Secure Sockets Layer / Transport Layer Security) ***(LAMARA, 2018).***

3.5.4 Déroulement de l'établissement de connexion d'un switch OpenFlow au contrôleur

Le déroulement de l'établissement de connexion d'un switch OpenFlow au contrôleur est résumé par les trois cas suivants :

- **Cas n°1**

Tout d'abord, il faut renseigner l'adresse IP du contrôleur au niveau du switch, il peut y en avoir plusieurs pour la redondance. Lors du démarrage du switch, le switch OpenFlow envoie un paquet *OFPT_HELLO* avec le numéro de version d'OpenFlow supporté. Le contrôleur vérifie la version d'OpenFlow supporté par le switch et lui répond par un *OFPT_HELLO* en indiquant la version d'OpenFlow avec laquelle ils communiqueront. La connexion est ainsi établie.

La figure (3.5) présente les principales étapes de la connexion entre le contrôleur et le commutateur OpenFlow.

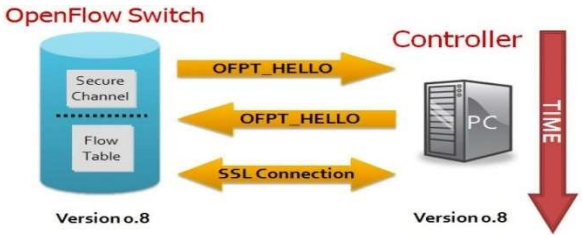


Figure 3.5:Connexion au contrôleur

- **Cas n°2**

Comme dans le cas précédent, le switch OpenFlow envoie son paquet OFPT_HELLO avec la version, le contrôleur s'aperçoit qu'il ne supporte pas la version OpenFlow du switch. Il lui retourne donc un paquet OFPT_ERROR en indiquant que c'est un problème de compatibilité, comme illustré dans la figure (3.6) :

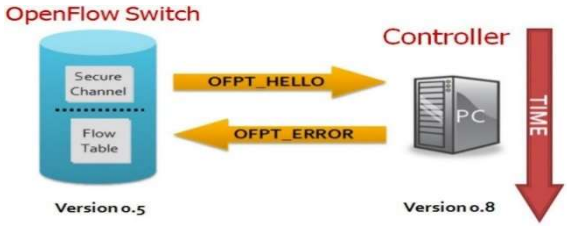


Figure 3.6:Versions différentes --> échec de connexion

- **Cas n°3**

Comme dans les cas précédents, le switch envoie un paquet OFPT_HELLO à son contrôleur, si celui-ci ne répond pas il tente alors de joindre les éventuels autres contrôleursqui lui ont été

paramétrés. S'il ne parvient pas à les joindre, il se met alors en mode "EMERGENCY MODE". Le switch utilise sa table de flux par défaut, si toutefois un paquetne correspond à aucun enregistrement dans la table il le supprime, comme le montre la figure ci-dessous ***(Wapiti, 2010)*** :

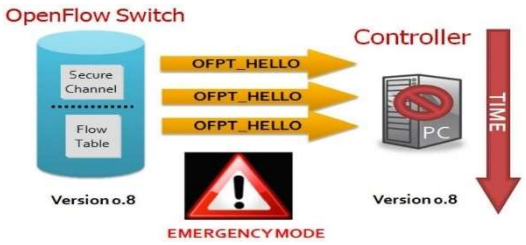


Figure 3.7:Controleur (s) injoignables (s)

3.5.4.1 Tables OpenFlow

Cette section décrit la structure de la table de flux (Flow Table) et celle de la table de groupe (Group Table).

1) Table de flux « Flow table »

Chaque table de flux dans le commutateur contient un ensemble d'entrées de flux ; chaque entrée de flux est constituée de champs de correspondance « Match Fields », compteurs, et un ensemble d'instructions à appliquer au paquet correspondant, comme indiqué dans la figure suivante :

Figure 3.8:Principaux champs d'une entrée d'une table de flux

2) Champs de Correspondance « Match Fields »

Utilisés par le contrôleur lors de la recherche de l'entrée correspondante au paquet, cela consiste à vérifier le port d'entrée ou l'entête du paquet afin d'y appliquer une action X. L'ensemble des champs du match Field (Le Tableau **(3.1)** illustre les champs pour la version 1.2 du protocole OpenFlow) :

Champ	N° couche	Application
Port d'entrée	2	Tous les paquets sur le port.
Adresse MAC source	2	Tous les paquets sur le port active.
Adresse MAC destination	2	Tous les paquets sur le port active.
Type Ethernet	2	Tous les paquets sur le port active.
ID VLAN	2	Tous les paquets avec un tag VLAN.
Priorité VLAN	2	Tous les paquets avec un tag VLAN.
Etiquette MPLS	2,5	Tous les paquets avec un tag MPLS.
Classe de trafic MPLS	2,5	Tous les paquets avec un tag MPLS.
Adresse IPv4 source	3	Tous les paquets IPv4 et ARP.
Adresse IPv4 destination	3	Tous les paquets IPv4 et ARP.
Protocole IPv4	3	Tous les paquets IPv4/6 sur Ethernet et ARP.
TOS bits IPv4	3	Tous les paquets IPv4.
Port de transport source	4	Tous les paquets UDP, TCP, SCMP et ICMP.
Port de transport destination	4	Tous les paquets UDP, TCP, SCMP et ICMP.

Tableau 3. 1: Champs de correspondance (Matching Fields) openflow

3) Compteurs

Servent essentiellement à garder des statistiques sur les flux pour décider si une entrée de flux est active ou non. Pour chaque table, chaque flux, chaque port, chaque file d'attenteet chaque groupe de tables, des compteurs de statistiques sont maintenus.

4) Instructions

Les instructions associées à une entrée de flux peuvent être soit des actions à appliquer au paquet correspondant soit des modifications du traitement (pipeline).

Les actions à appliquer sur un paquet peuvent être :

- ***Output :*** Envoi du paquet par un port spécifié.
- ***Set-field :*** Modifier la valeur d'un champ d'en-tête spécifique, comme le TTL (Time To Live), l'adresse MAC, ou l'ID du VLAN (Virtual Local Area Network),etc…
- ***Drop :*** Supprimer le paquet.
- ***Group :*** Traiter le paquet selon un groupe spécifique (entrée spécifique de la tablede groupe).

Tandis que les instructions de modification du pipeline peuvent être :

- ***Apply-Actions :*** Pour appliquer immédiatement les actions sur le paquet.
- ***Clear-Actions :*** Pour supprimer une liste des actions du paquet.
- ***Write-Actions :*** Ajouter une liste d'actions au paquet.
- ***Write-Metadata :*** Ajouter des données utiles pour le séquencement entre lestables OpenFlow.
- ***Goto-Table :*** Indique que le paquet doit être acheminé vers une table d'indicesupérieur.

Une entrée de table de flux est identifiée par ses champs de correspondance et sa priorité : les champs de correspondance et la priorité combines identifient une entrée de flux unique dans la table de flux. L'entrée de flux qui caractérise tous les champs et a une priorité qui égale a 0 est appelée entrée de flux manquée dans la table de flux. Le schéma de la figure **(3.9)** résume le traitement d'un paquet dans un réseau OpenFlow :

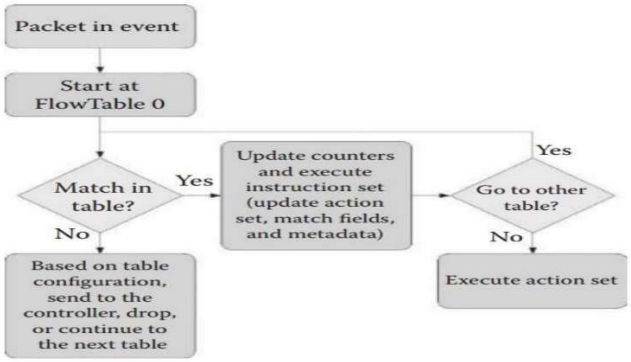


Figure 3.9:Etapes de traitement des paquets dans un réseau openflow

1.1.1 Table de groupe « Group table »

Une table de group « Group table » se compose d'un ensemble d'entrées de groupe. Ceci permet à OpenFlow d'envisager d'autres méthodes de transfert comme l'application d'un ensemble de transactions d'actions sur un type de flux. La figure suivante montre la structure d'une entrée de groupes :

Figure 3.10:Les principaux champs d'une entrée de table de groupe

Une entrée de groupe se compose de :

- ***Identifiant de groupe :*** un entier sur 32 bits identifie de manière unique legroupe.
- ***Type de groupe :*** pour déterminer la sémantique du groupe.
- ***Compteur :*** Ensemble des compteurs qui vont être mise à jour une fois qu'unpaquet est traité par le groupe.
- ***Action Buckets :*** Contient une liste ordonnée de conteneurs d'actions, où chaqueconteneur

contient une série d'actions à exécuter ***(IDOUDI, 2014)***.

3.6 Les messages openflow

Tous les messages sont initiés par l'entête OpenFlow qui définit la longueur, l'identificateur de transaction et le type de message. Connaissant l'adresse IP du Contrôleur,le commutateur initie une connexion TLS avec le commutateur, ces messages sont sécuriséset assurés par une connexion TLS sur TCP. On trouve trois catégories de message (symétrique, asymétrique, et Contrôleur-commutateur).

3.6.1 Messages Contrôleur-Commutateur

Ces messages servent à vérifier l'état du switch, gérer et vérifier son état et peuventdemander des réponses de la part du switch. Ces messages sont initiés par le Contrôleur.

- ***Features :*** le contrôleur peut demander l'identité et les capacités d'un switch celase passe en envoyant une requête.
- ***Modify-state :*** ce type de message permet de gérer l'état des switches ; ilspermettent de modifier ou ajouter, effacer les entrées dans les tables openflow.
- ***Configuration :*** le contrôleur peut définir les paramètres de configuration duswitch.
  Ce dernier, répond aux requetés envoyés par le contrôleur :

- ***Read-state :*** le contrôleur utilise ces messages pour collecter différentes informations du switch, tel que les statistiques, les capacités et sa configuration actuelle.
- ***Packet-out :*** ces messages sont utilisés pour le transfert de paquets reçus par les messages Packet-in. Le message doit contenir un paquet entier ou l'ID du bufferfaisant référence à un paquet stocké dans le switch. Si la liste d'actions du message est vide le paquet sera détruit.
  Barrier : utilisé par le contrôleur pour recevoir des notifications de l'opération terminée. La figure suivante présente les échanges de la connexion entre le contrôleur et le commutateur OpenFlow

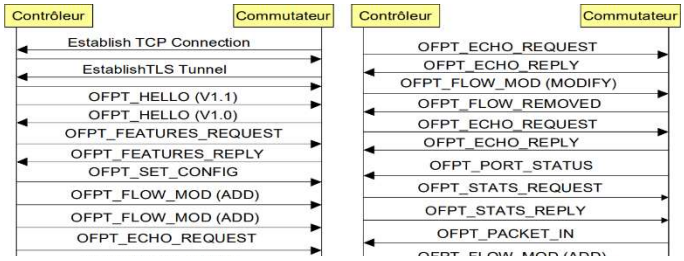


Figure 3.11:Echanges openflow entre contrôleur et commutateur

3.6.2 Messages asynchrones

Les messages asynchrones sont initiés par le commutateur, et ils sont utilisés pour mettre à jour le contrôleur des événements du réseau et des changements de l'état du commutateur.

Les commutateurs envoient des messages asynchrones au contrôleur afin de signaler l'arrivée d'un paquet, un changement d'état du commutateur ou une erreur.

Il existe sept messages asynchrones principaux, à savoir :

- ***Packet-in :*** avec ce message le switch transfert le contrôle du paquet aucontrôleur.
- ***Flow-removed :*** le switch informe le contrôleur de la suppression d'une entréedans la table du flux.
- ***Port-satus :*** avec ce message le switch informe le contrôleur d'un changementsur un port.
- ***Role-status :*** quand un nouveau contrôleur est aux commandes, le switch envoiun Role-status à son contrôleur.
- ***Controller-status :*** Ce message est généré lorsque le statut du contrôleur estmodifié.
- ***Table-status :*** Il est indiqué par le message OFPT_TABLE_STATUS et il est envoyé au contrôleur lorsqu'il y a un changement dans l'état de la table.
- ***Request-forward :*** Il est utilisé par un contrôleur pour informer les autrescontrôleurs d'une modification de l'état des groupes et des compteurs.

3.6.3 Messages symétriques

Les messages symétriques sont envoyés sans aucune sollicitation ni du switch ni du contrôleur. Ces messages sont :

- ***Hello :*** envoyer pour vérifier la connectivité entre le contrôleur et le switch.
- ***Echo :*** une fois la connexion établie, ces messages sont échangés entre lecontrôleur et le switch. Chaque message ECHO_REQUEST doit être acquitté parun ECHO_REPLY.
- ***Error :*** utilisé par le contrôleur et le switch pour signaler un problème de connexion.
- ***Expérimenter :*** Ce message est identifié par le champ de l'expérimentateur et il estutilisé pour créer une nouvelle API et gérer des objets complètement nouveaux. Ils ne sont pas attribués à un objet Open Flow spécifique ***(Beroual, 2020)***.

3.7 Quelques contrôleurs SDN

Plusieurs Contrôleurs SDN existent actuellement (POX, NOS, Ryu, Open Floodlight). Un

Contrôleurs est un logicielle qui contrôle les équipements réseau comme un système d'exploitation.

Le Contrôleur, est programmable avec un langage de programmation moderne (Java, en Python ou C++). La majorité des Contrôleurs actuels et ceux les plus utilisés sont programmables en Python ; En programmant le Contrôleur, l'utilisateur peut redéfinir la politique du réseau sans reconfigurer les équipements Switches / Routeurs.

Les applications peuvent interagir avec le contrôleur pour accéder au réseau à travers l'API proposée par le contrôleur. Cette API est appelée aussi Northbound Interface (Interface vers le Nord).

3.7.1 Opendaylight

Il s'agit d'une plate-forme pour le Software Defined Network (SDN). Son rôle est de fournir le contrôle centralise et programmable ainsi que les protocoles open-source de surveillance des périphériques réseau.

En raison de la nature open-source du contrôleur, il permet aux utilisateurs de minimiserla complexité opérationnelle et donc de prolonger la dure de vie de leur infrastructure existante. Comme beaucoup d'autre contrôleur SDN, tel que Nox, Pox …etc. OpenDaylightprend en charge openflow et offre des solutions réseau prêtes à installer dans le Framework de sa plate-forme.

La communauté OpenDaylight est en train de développer une architecture SDN qui prenden charge un large éventail de protocoles et peut évoluer rapidement dans la direction du SDN, sans se baser sur les objectifs d'un seul fournisseur.

3.7.2 Nox

NOX a été le premier contrôleur OpenFlow écrit en C++ et il fournit également une APIpour Python. Il a servi de base à de nombreux projets de recherche et développement dans les premières explorations de l'espace OpenFlow et SDN.

Il sert de plate-forme de contrôle de réseau, qui fournit une interface de programmation de haut niveau pour la gestion et le développement d'applications de contrôle de réseau. Ses abstractions à l'échelle du système transforment la mise en réseau en un problème logiciel.

3.7.3 Pox

POX fournit un cadre pour communiquer avec les commutateurs SDN à l'aide du protocole OpenFlow ou OVSDB. Les développeurs peuvent utiliser POX pour créer un contrôleur SDN à l'aide du langage de programmation Python. C'est un outil populaire pour l'enseignement et la recherche sur les réseaux définis par logiciel et la programmation d'applications réseau.

3.7.4 Beacon

Beacon est un contrôleur OpenFlow développé en Java et publié en 2010. Il a été utilisédans plusieurs projets de recherche, considéré comme plus approprié pour les réseaux d'entreprise et de centres de données.

Aujourd'hui, Beacon est considéré comme inactif, Il utilise une approche statique dans laquelle un nombre fixe de commutateurs sont affectés à un thread de travail, ce contrôleura également été utilisé dans d'autres projets tels que Floodlight ou Opendaylight.

3.7.5 Floodlight

Il est sous licence Apache et il est l'une des contributions importantes de Réseaux Big Switch à la communauté open source écrit en java. Floodlight est l'un des contrôleurs SDNles plus populaires prenant en charge les commutateurs physiques et virtuels compatibles OpenFlow. Il est basé sur un contrôleur de bacon de l'Université de Stanford. L'architecture Floodlight est modulaire ***(Meridji, 2019).***

Le tableau **(3.2)** montre la comparaison basée sur les fonctionnalités des contrôleurs SDN open source les plus populaires ***(Beroual, 2020)***.

Tableau 3. 2:Comparaison des contrôleurs SDN open source les plus populaires

Caractéristiques	Floodlight	Opendaylight	Nox	Pox	Beacon
Développe par	Big switch Networks	Linux Fondation	Nicira	Nicira	
Ecrit en langage	Java	Java	C++, Python	Python	Java
Langages Supporte	Java, python	Java	C++, Python	Python	Java
Open source	Oui	Oui	Oui	Oui	Oui
Interface utilisateur	Web	Web		Web	Web
Virtualisation	Mininet, openVswitch	Mininet, OpenVswitch	Mininet, OpenVswitch	Mininet, OpenVswitch	Mininet, OpenVswitch
Interface	South Bound (OpenFlow), North Bound (Java, REST).	South Bound (OpenFlow et autres protocoles), North Bound (Java RPC).	South Bound (OpenFlow)	South Bound (OpenFlow).	South Bound (OpenFlow)
Plateforme	Linux, mac, Windows.	Linux, Windows.	Linux	Linux	Linux, MAC OS, And Windows.
Documentation	Site officiel	Moyen	Un peu difficile à trouver.	Moyen	Équitable

3.8 Conclusion

Dans ce chapitre, nous avons donné une base théorique sur les réseaux définis par logiciels (SDN), dans lequel nous avons fourni une vue générale sur SDN (Software Defined Networking), notamment les avantages. Ensuite nous avons présentéle protocole OpenFlow, son fonctionnement, ainsi que quelques contrôleurs SDN. Enfin nous avonsprésenté quelques défis de SDN, et les solutions récentes proposées pour surmonter ces défis.

Dans la suite de notre mémoire nous allons investiguer les possibilités de mise en œuvre d'une solution SDN à partir d'un réseau classique afin de mettre cette technologie en temps réel.

Chapitre 4

Evaluation des Performances de POX et RYU

Chapitre 4 Evaluation des performances de POX et RYU

4 Introduction

Après avoir abordé les aspects théoriques de la technologie SDN, nous arrivons maintenant à la partie applicative où nous allons émuler la mise en place de ces solutions sur une topologie réelle.

Dans ce qui suit, nous allons présenter l'un des outils de simulation le plus connu et utilisé dans le domaine de simulation du SDN.

Il y a plusieurs outils de simulation de systèmes distribués. Les plus connus sont Packet Tracer, Mininet, Cloud Sim, GNS3…. Nous avons présenté le Framework Mininet, qui permet la simulation de SDN.

Ce chapitre représente la première partie de notre travail, nous commençons d'abord par une présentation de Mininet, puis nous parlerons du contrôleur POX et RYU ainsi que leurs fonctionnalités.

4.1 Préparation de l'environnement de test SDN

L'expérience est réalisée dans un environnement virtuel, tous les appareils étant émulés. Le matériel réel utilisé pour réaliser cette expérience était un ordinateur portable HUAWEI et ASUS équipé d'un processeur i7 8550U CPU et de 8 Go de RAM et i7 7500 U CPU et de 16 Go de RAM.

Cette expérience est émulée dans Virtual Box (logiciels de virtualisation) sous système d'exploitation Ubuntu-20.04, ce système comprend Pox- v2.0, RYU- v2.0 et Mininet- v2.3.0.

4.2 Mininet

Mininet est un *émulateur de réseau* qui crée un réseau d'hôtes virtuels, de commutateurs, de contrôleurs et de liens. Les hôtes Mininet exécutent un logiciel réseau Linux standard et ses commutateurs prennent en charge OpenFlow pour un routage personnalisé très flexible et une mise en réseau définie par logiciel.

Mininet prend en charge la recherche, le développement, l'apprentissage, le prototypage, les tests, le débogage et toute autre tâche qui peut bénéficier d'un réseau expérimental complet sur un ordinateur portable ou un autre PC.

Mininet :

- Fournit un banc d'essai réseau simple et peu coûteux pour le développement d'applications OpenFlow.
- Permet à plusieurs développeurs simultanés de travailler indépendamment sur la même topologie.
- Prend en charge les tests de régression au niveau du système, qui sont reproductibles et facilement empaquetés.
- Permet des tests de topologie complexes, sans avoir besoin de câbler un réseau physique.
- Inclut une CLI qui prend en charge la topologie et OpenFlow, pour le débogage ou l'exécution de tests à l'échelle du réseau.
- Prend en charge les topologies personnalisées arbitraires et inclut un ensemble de base de topologies paramétrées, est utilisable directement sans programmation, mais fournit également une API Python simple et extensible pour la création et l'expérimentation de réseaux.

Mininet offre un moyen simple d'obtenir un comportement correct du système et d'expérimenter des topologies.

Les réseaux Mininet exécutent du code réel, y compris des applications réseau Unix/Linux standard, ainsi que le véritable noyau Linux et la pile réseau (y compris toutes les extensions de noyau dont il dispose, tant qu'elles sont compatibles avec les espaces de noms réseau.)
Pour cette raison, le code qu'est développé et tester sur Mininet, pour un contrôleur OpenFlow, un commutateur modifié ou un hôte, peut être déplacé vers un système réel avec des modifications minimes, pour des tests en conditions réelles, une évaluation des performances et un déploiement. Surtout, cela signifie qu'une conception qui fonctionne dans Mininet peut généralement passer directement aux commutateurs matériels pour le transfert de paquets à débit de ligne ***(Mininet, 2022)***.

4.3 Open V switch

Open Vswitch, parfois abrégé en OVS, est une implémentation open source d'un commutateur multicouche virtuel distribué et de qualité de production sous licence open source Apache 2.0.

Il est conçu pour permettre une automatisation massive du réseau via une extension programmatique, tout en prenant en charge les interfaces et protocoles de gestion standard (par

exemple, NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag).

En outre, il est conçu pour prendre en charge la distribution sur plusieurs serveurs physiques similaires au vswitch distribué vNetwork de VMware ou au Nexus 1000V de Cisco ***(open v switch, 2016).***

4.3.1 Fonctionnement de open V switch

Le programme ovs-vsctl configure ovs-vswitchd en fournissant une interface de haut niveau avec sa base de données de configuration.

Ovs-vsctl se connecte à un processus ovsdb-server qui maintient un Open Base de données de configuration vSwitch. A l'aide de cette connexion, il interroge et applique éventuellement des modifications à la base de données, selon les commandes. Ensuite, s'il a appliqué des modifications, par défaut, il attend jusqu'à ce que ovs-vswitchd a fini de se reconfigurer avant de se fermer. (Si on utilise ovs-vsctl lorsque ovs-vswitchd n'est pas en cours d'exécution, utilise --no-wait.) ovs-vsctl peut exécuter n'importe quel nombre de commandes en une seule exécution, comme une seule transaction atomique sur la base de données ***(Openvswitch,2022)***.

4.3.2 Interagir avec open V switch

Voici une liste de certaines des commandes qui peuvent être utilisées dans ce programme :

- ***Ovs-vsctl show :*** Aperçu de la configuration actuel.
- ***Ovs-vsctl add-br <pont> :*** Création d'un nouveau bridge.
- ***Ovs-vsctl liste-br :*** Obtenir la liste des bridges.
- ***Ovs-vsctl add-port <pont><port> :*** Patch entre deux bridges.
- ***Ovs-vsctl list-ports <pont> :*** Obtenir la liste des enregistrements d'une table sélectionnée.
- ***Ovs-vsctl set port eth3 vlan-mode=Access tag=982 :*** Ajouter un vlan un tag sur un port.
- ***Ovs-vsctl remove port eth3 tag 982 :*** Supprimer un vlan d'un port.

4.4 Contrôleurs SDN

Dans notre expérience, on a choisi deux contrôleurs de SDN parmi plusieurs contrôleurs existe sur les marches, qui sont les suivants :

4.4.1 Pox :

POX est un contrôleur OpenFlow/Software Defined Networking (SDN) open source basé sur Python. POX est utilisé pour accélérer le développement et le prototypage de nouvelles applications réseau. Le contrôleur POX est préinstallé avec la machine virtuelle Mininet.

Il permet d'exécuter facilement des expériences OpenFlow/SDN. POX peut être passer à différents paramètres selon des topologies réelles ou expérimentales, il permet ainsi d'exécuter des expériences sur du matériel réel, des bancs d'essai ou dans un émulateur Mininet ***(Kaur, 2014).***

4.4.1.1 Fonctionnalité de contrôleur POX

Les composants POX sont des programmes Python supplémentaires qui peuvent être appelés lorsque POX est démarré à partir de la ligne de commande. Ces composants implémentent la fonctionnalité réseau dans le réseau défini par logiciel. POX est livré avec certains composants de stock déjà disponibles.

Les composants du stock POX sont documentés dans le wiki POX et le code de chaque composant se trouve dans le répertoire ~/pox/pox sur l'image de la machine virtuelle Mininet 2.3.

4.4.2 RYU

Ryu Controller est un contrôleur de réseau défini par logiciel (SDN) ouvert conçu pour augmenter l'agilité du réseau en facilitant la gestion et l'adaptation de la gestion du trafic. En général, le contrôleur SDN est le cerveau de l'environnement SDN, communiquant des informations jusqu'aux commutateurs et routeurs avec les API sud, et jusqu'aux applications et à la logique métier avec les API nord. Le contrôleur Ryu est pris en charge par NTT et est également déployé dans les centres de données cloud NTT.

Le contrôleur Ryu fournit des composants logiciels, avec des interfaces de programme d'application (API) bien définies, qui permettent aux développeurs de créer facilement de nouvelles applications de gestion et de contrôle de réseau. Cette approche par composants aide les organisations à personnaliser les déploiements pour répondre à leurs besoins spécifiques ***(Studios SDxCentral, 2016).***

4.4.2.1 Fonctionnement de contrôleur RYU

La gestion centrale des applications Ryu.

- Charger les applications Ryu
- Fournir des contextes aux applications Ryu
- Acheminer les messages entre les applications Ryu

4.5 Configuration et émulation

4.5.1 Installation et configuration Mininet

L'installation et la configuration de Mininet se fait sur Ubuntu, les étapes de configuration peuvent être résumées dans les points suivants :

- Pour installer le package de base '*Mininet'*, il faut utiliser la commande présentée dans la figure **(4.1)** :>sudo apt-get Install Mininet

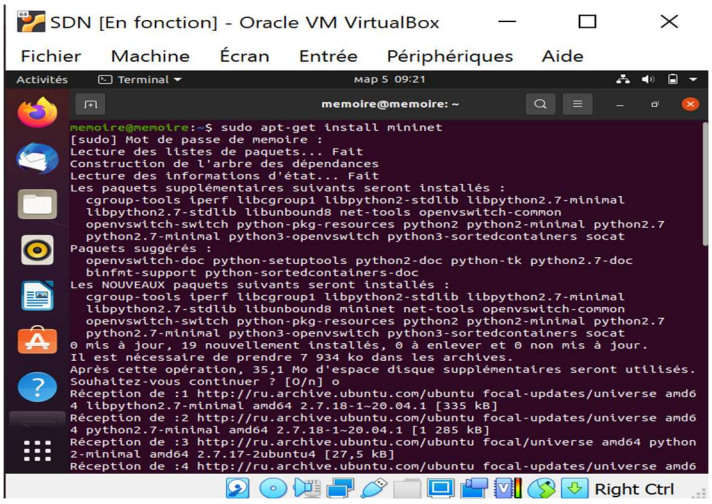


Figure 4.1:Installer Mininet

- Pour installer le package *'virtualbox-guest-x11',* nous utilisons la commande suivante : *>sudo apt-get Install virtualbox-guesr-x1*

  Comme Ilustre la figure **(4.2).**

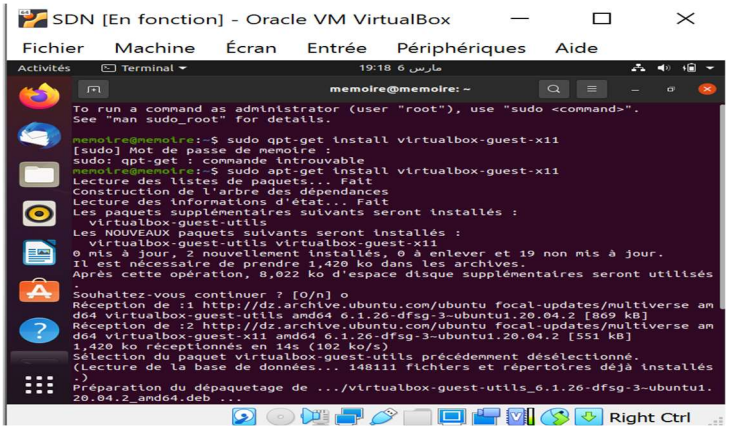


Figure 4.2: Installer le package VirtualBox

- Pour consulter la version Mininet nous utilisons la commande suivante :

>mn –version

- Mininet prend en charge plusieurs commutateurs et contrôleurs OpenFlow. Pour ce test, utiliserons Open Vswitch en mode bridge/standalone.

>sudo mn –switch ovsbr –test pingall comme montre la figure **(4.3).**

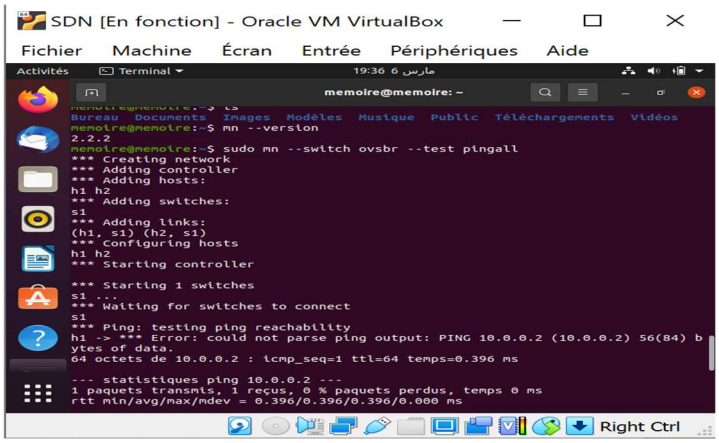


Figure 4.3: La version de Mininet et test de OVS

4.5.2 Installation de open V switch

Mininet se plaint qu'Open Vswitch ne fonctionne pas, donc il faut installer open Vswitch.

- Nous utilisons la commande suivante :

>sudo Apt Install openvswitch-switch openvswitch-Common comme illustrer la figure **(4.4)**

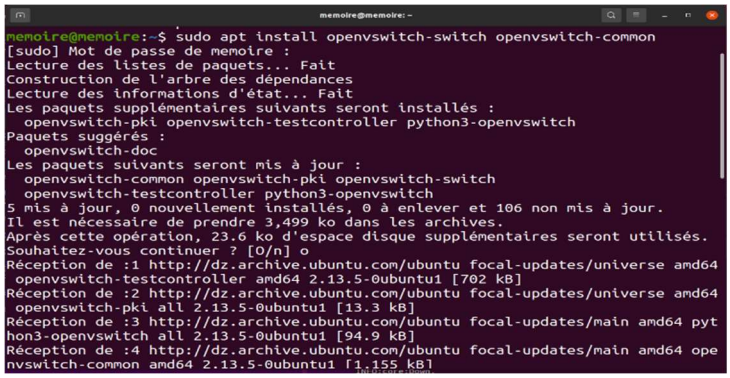


Figure 4.4: Installation de open V switch

La figure **(4.5)** montre comment faire pour vérifier la version de python.

>echo py sys.version | sudo mn -v output

Figure 4.5: La version python

4.5.3 Installer RYU

Exécute la commande Install avec l'indicateur -y pour installer rapidement les packages et les dépendances de contrôleur RYU.

>sudo apt-get update -y

>sudo apt-get Install -y python3-ryu

4.6 Fonctionnement de base de Mininet

Topologie Minimal : il s'agit de la topologie par défaut avec un contrôleur, un commutateur et 2 hôtes, dans la figure **(4.6)** nous avons utilisons une seule commande qu'est **#sudo mn** pour créer la topologie minimale.

Figure 4.6:Topologie minimale

Lorsque nous démarrons automatiquement l'émulateur Mininet, Mininet nous donnerons une topologie avec 2 hôtes, un contrôleur SDN et un Open Vswitch, puis nous verrons la ligne de Commande Mininet *"Mininet>"* agir comme un terminal sur le contrôleur SDN pour afficher et configurer tout le nœud dans la topologie Mininet.

En parallèle nous ouvrons une autre fenêtre pour lancer le contrôleur POX en utilisant la commande suivante, comme illustrer la figure **(4.7).**

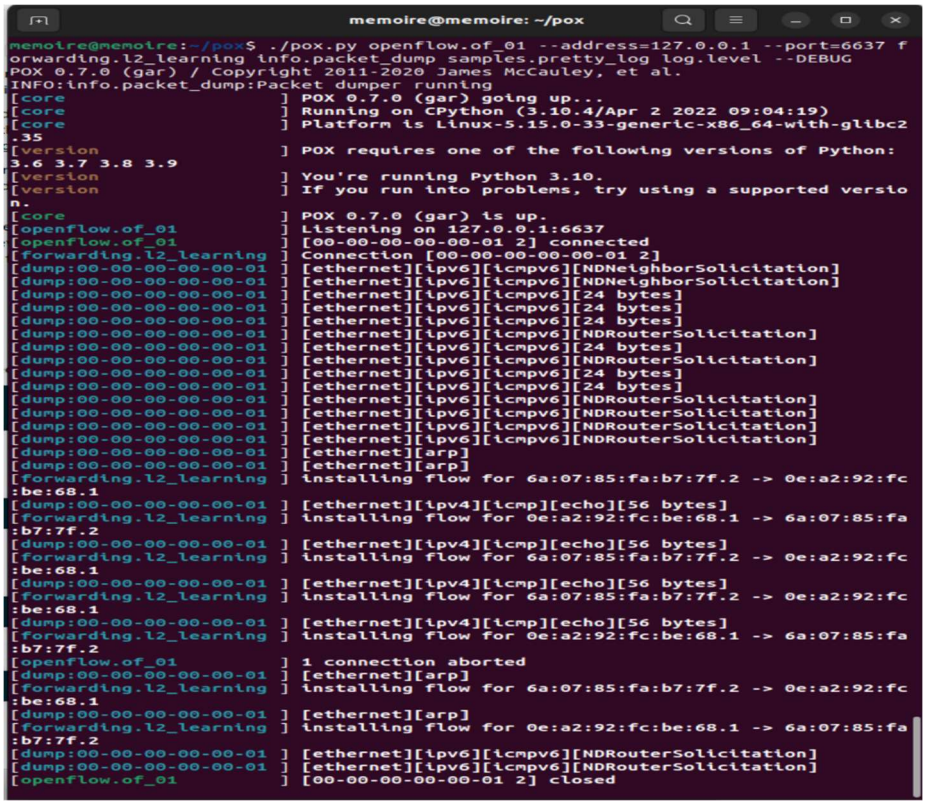


Figure 4.7: Lancement de contrôleur POX

Notre contrôleur est bien lancé. Quand en sortant de Mininet en utilisant ‘exit ‘ et le contrôleur ‘Pox’ est fermé ‘closed’ automatiquement. Comme illustrer la figure ***(4.8).***

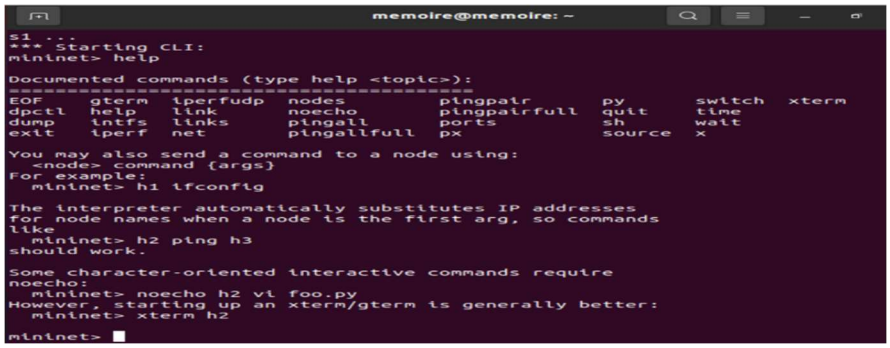


Figure 4.8: La commande help

Pour connaître la commande de base sur le terminal Mininet, nous pouvons faire help command "Mininet> help".

Parce qu'il s'agit d'une base de ligne de commande, il est peut-être difficile de comprendre à quoi ressemble notre topologie, nous pouvons donc utiliser une commande pour comprendre la topologie du Mininet et comprendre comment ils se sont connectés.

- ***Mininet> net*** -la figure ***(4.9)*** montre comment Affiche la topologie du réseau. Quelle machine est connectée à quel hôte et le nom de l'interface réseau de connexion.

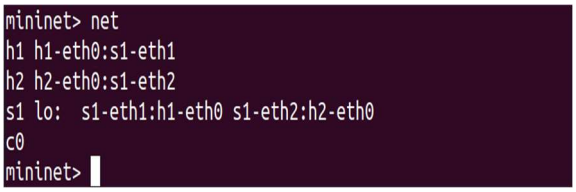


Figure 4.9: La commande net

- ***Mininet>nœuds-*** la figure ***(4.10)*** montre comment faire Pour voir Nœud disponible sur la topologie.

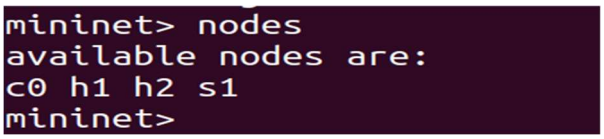


Figure 4.10: La commande Nœuds

- ***Mininet> dump*** – la figure ***(4.11)*** illustrer comment Affiche les adresses IP de chaque machine ainsi que le nom de la carte réseau.

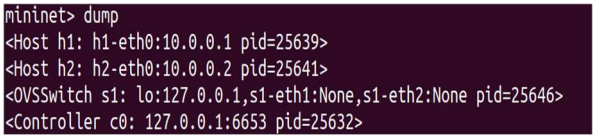


Figure 4.11: La commande dump

- ***Mininet>links-*** la figure ***(4.12)*** illustrer comment faire Pour voir les liens disponibles pour interconnecter tous les nœuds sur la commande Mininet de topologie utilisée.

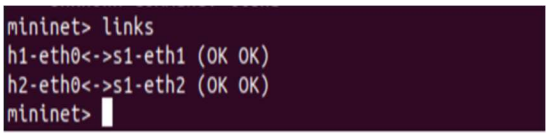


Figure 4.12: La commande links

- ***Mininet> pingall –*** la figure ***(4.13)*** illustrer comment Permet de tester la connectivité du réseau. Toutes les machines vont se ping entre elles.

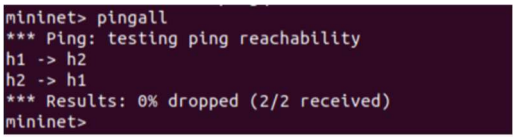


Figure 4.13: La commande pingall

- ***Mininet > h1 ping h2 –*** la figure ***(4.14)*** montre comment Demande à l'hôte h1 d'effectuer un ping sur l'hôte h2.

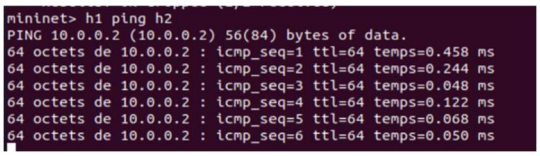


Figure 4.14: La commande h1 ping h2

- ***Mininet> h1 ifconfig -a –*** la figure ***(4.15)*** montre comment Effectue et affiche les résultats de la commande *if config* sur la machine souhaitée.

Figure 4.15: La commande h1 ifconfig -a

- ***Mininet> exit-*** la figure ***(4.16)*** montre comment faire pour sortir de *'mininet>'*.

Figure 4.16: La commande exit

- ***# sudo mn -c :*** la figure **(4.17)** montre comment Effectuez une installation propre du Mininet.

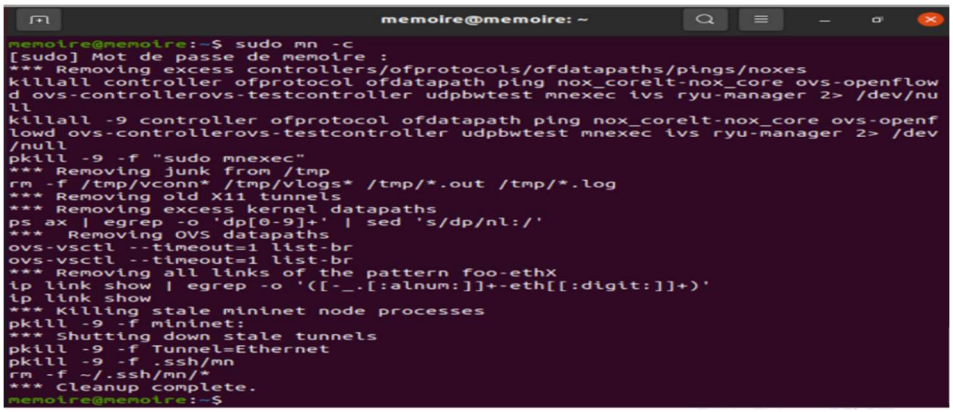


Figure 4.17: La commande sudo mn -c

4.7 Les différents types de topologie

Mininet crée un **réseau virtuel réaliste,** exécutant **un noyau, un commutateur et un code d'application réels,** sur une seule machine (VM, cloud ou native), en quelques secondes ***(Mininet, 2022)***.

Topo= TOPO représente la topologie du réseau virtuel, où topo pourrait être :

Single, x : c'est une topologie avec un contrôleur, un seul commutateur avec X hosts qui sont attachés à lui. Les figures **(4.18) (4.19)** explique comment nous avons créé la topologie single Avec la commande suivante : >sudo mn –topo single, 10 –controller=remote

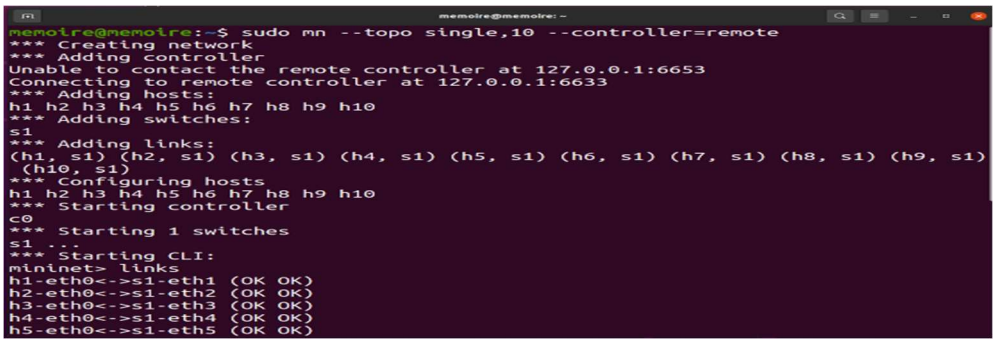


Figure 4.18: La topologie single

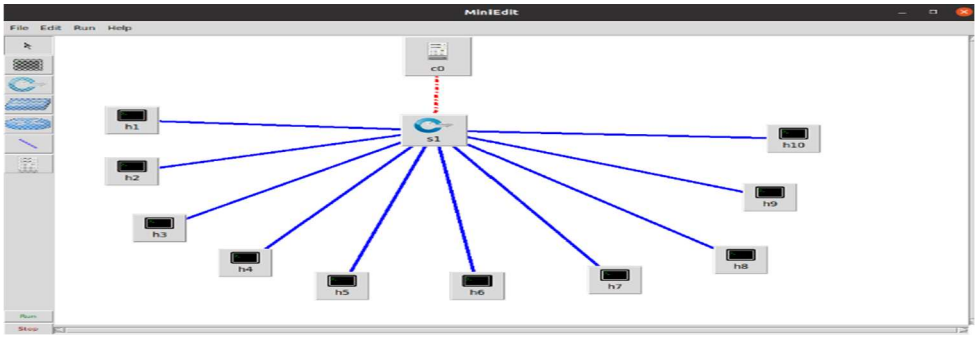


Figure 4.19: La topologie single sur le miniedit

Linear, X : crée X commutateurs connectés de manière linéaire/en guirlande, chaque commutateur avec un hôte connecté, en utilisant la commande >sudo mn --topo linear,10 --

controller=remote, comme illustrer dans les figures **(4.20) (4.21).**

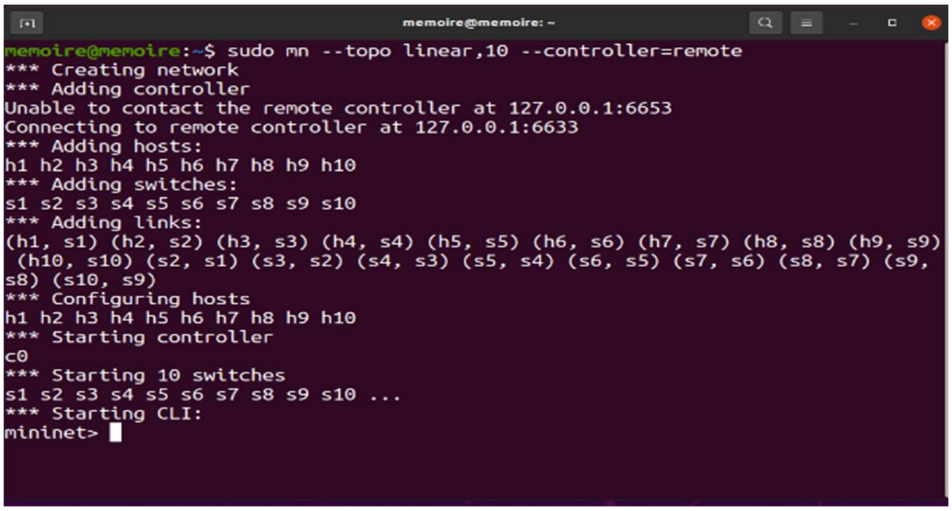


Figure 4.20: Topologie linéaire

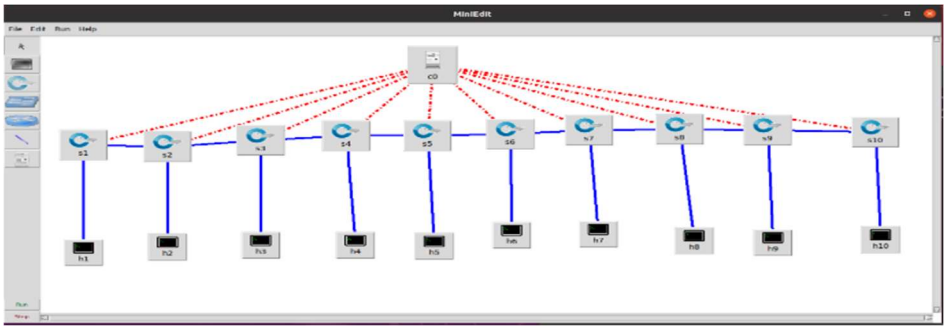


Figure 4.21: Topologie linéaire sur le miniedit

Tree : Une topologie arborescente est un type particulier de structure où de nombreux éléments connectés sont disposés comme les branches d'un arbre, en utilisant la commande suivante : >sudo mn --topo tree,3 --controller=remote, quest presenter dans les figures **(4.22) (4.23)**.

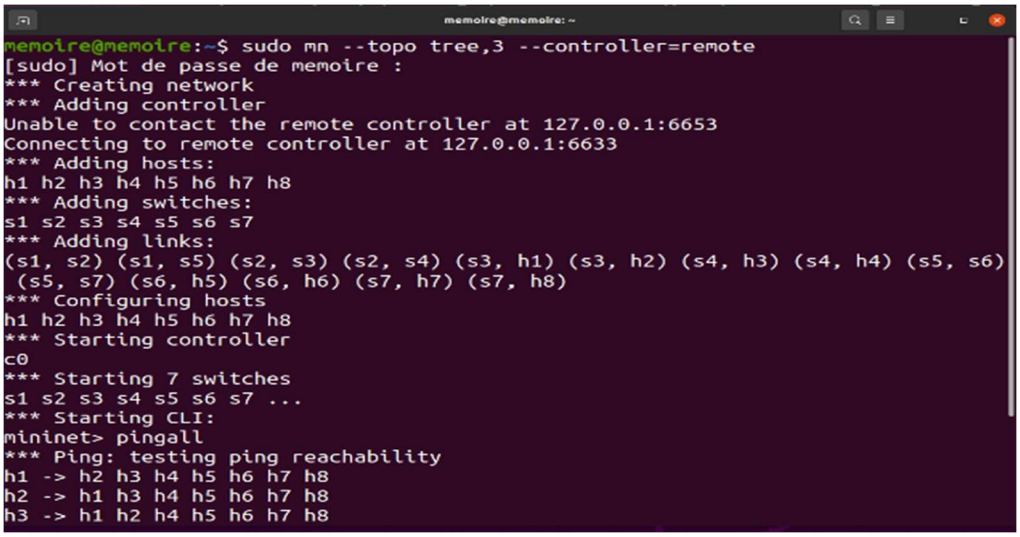


Figure 4.22: Topologie tree

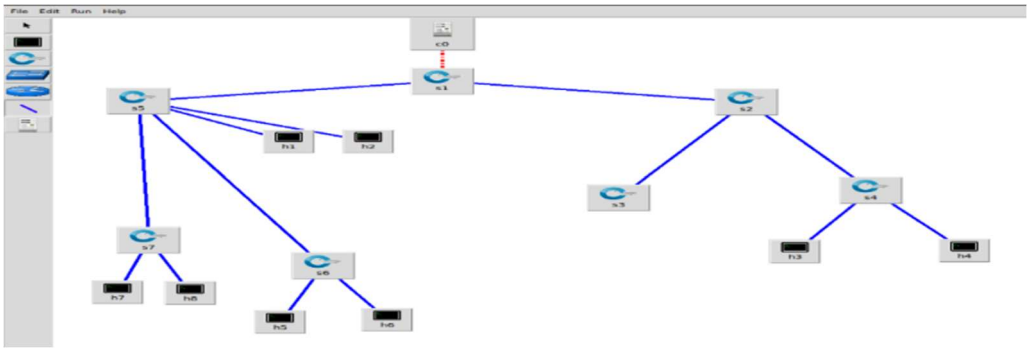


Figure 4.23: Topologie tree sur le miniedit

4.8 Création d'une topologie personnalise

La création de la topologie peut se faire en deux manières, en utilisant un outil intégré à Mininet sous le nom « MiniEdit » qui se repose sur une interface graphique, ou en développant un code source qui génère la topologie à son exécution ; nous avons choisi d'utiliser les deux méthodes.

4.8.1 Méthode 1 : MiniEdit

Mininet fournit un outil qui permet de créer sa propre topologie de manière graphique du nom de miniedit.

Pour la première des choses nous ouvrons les dossiers suivants (mininet, examples), dans une autre fenêtre on ouvre le dossier 'Pox', comme la figure **(4.24)** illustre :

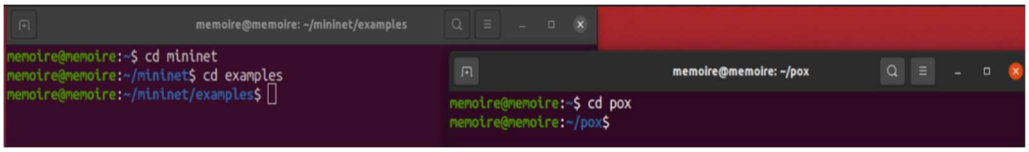


Figure 4.24:Pour démarrer notre création

Il faut d'abord lancer le contrôleur 'Pox', avec la commande suivante :

~Pox$ python3 pox.py forwarding.l2-learning ,comme illustre la figure **(4.25)**

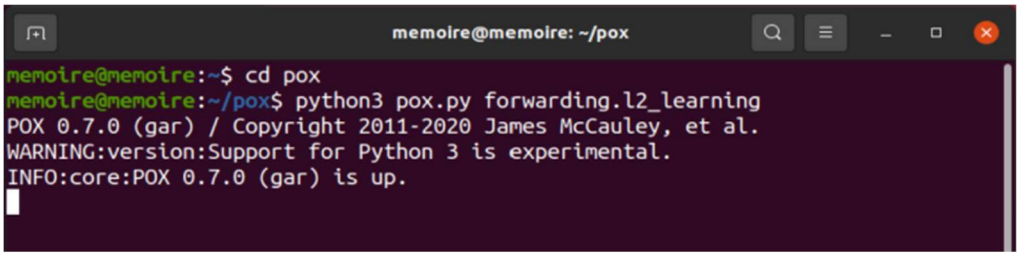


Figure 4.25:Lancement de contrôleur POX

Puis en exécutant notre interface en utilisant la commande suivante :

>~/mininet/examples$ sudo puthon3 miniedit.py où

>~/mininet/examples$python3./miniedit.py, comme illustre la figure **(4.26)**

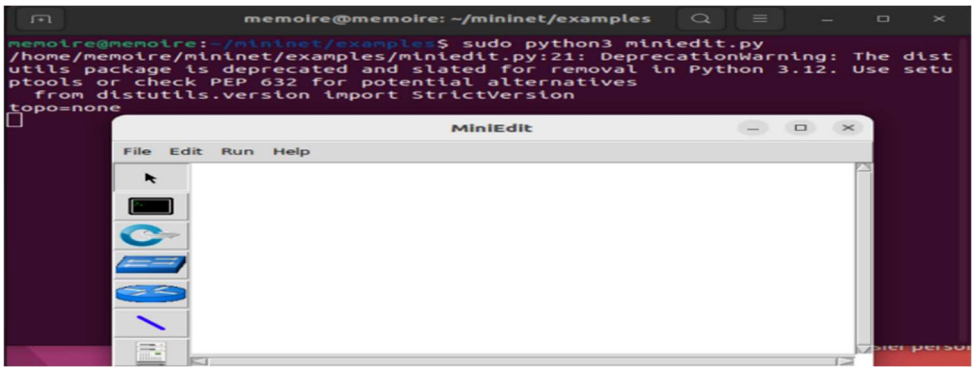


Figure 4.26:Lancement de miniedit

MiniEdit a une interface utilisateur simple qui présente un canevas avec une rangée d'icônes d'outils sur le côté gauche de la fenêtre et une barre de menus en haut de la fenêtre. Comme illustre la figure **(4.27).**

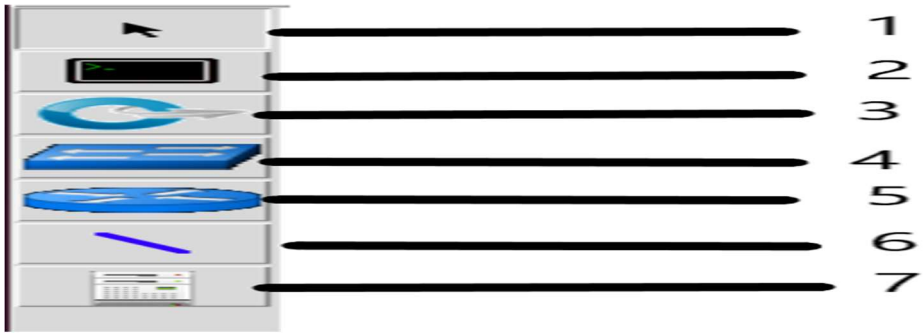


Figure 4.27: Icones

Les icônes représentent les outils suivants Dans l'ordre d'affichage de la figure précédente :

1. Permet de déplacer les objets dans la topologie.
2. Permet d'ajouter un hôte à chaque clic.
3. Permet d'ajouter un switch à chaque clic.
4. Permet d'ajouter un switch legacy à chaque clic.
5. Permet d'ajouter un routeur legacy à chaque clic.
6. Permet de faire des liens entre les objets de la topologie. Le clic doit être maintenu.
7. Permet d'ajouter un contrôleur à chaque clic.

Lorsque la topologie a été créée avec miniedit, il est possible de l'exporter dans un fichier Python en allant dans *File / Export Level 2 Script*. Le fichier Python pourra être utilisé comme un fichier Python normal. Le code python offre toutes les fonctionnalités que l'on peut trouver dans le client Mininet.

>sudo python3 final.py comme illustrer la figure **(4.28)**

```
memoire@memoire:~/mininet/examples$ sudo python3 final.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet>
```

Figure 4.28:Topologie personnaliser

Le scénario de simulation *'Exécuter'*, *'démarre'* Mininet actuellement affiché dans le canevas MiniEdit. Le bouton Stop. Lorsque la simulation MininEdit est à l'état "Exécuter", un clic droit sur les éléments du réseau révèle des fonctions opérationnelles telles que l'ouverture d'une fenêtre de terminal, l'affichage de la configuration du commutateur ou la définition de l'état d'un lien sur "up" ou "down".

La topologie du réseau utilisée dans cette expérience qui compose de 43 hôtes, 19 commutateurs et 3 contrôleurs. Comme présenter dans la figure **(4.29).**

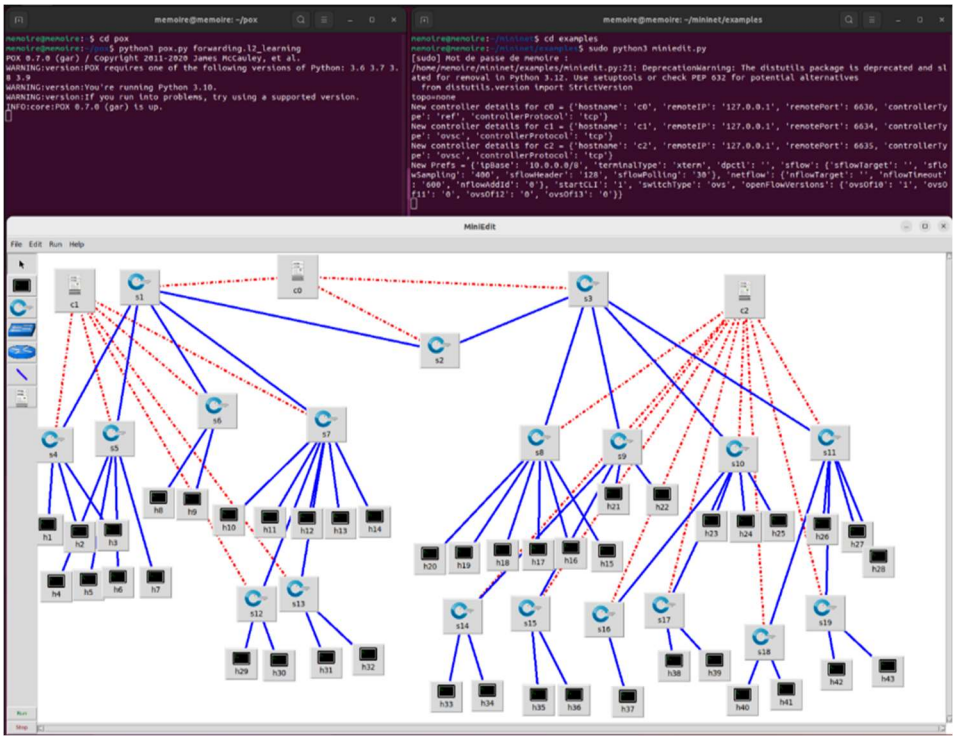


Figure 4.29: Création d'une topologie personnaliser

Nous avons pu créer cette topologie de réseau personnalisée complexe en quelques minutes à l'aide de MiniEdit. L'écriture manuelle d'un script de topologie personnalisée Mininet pour créer ce scénario aurait pris beaucoup plus de temps. Comme illustrer les figures **(4.30) (4.31) (4.32) (4.33).**

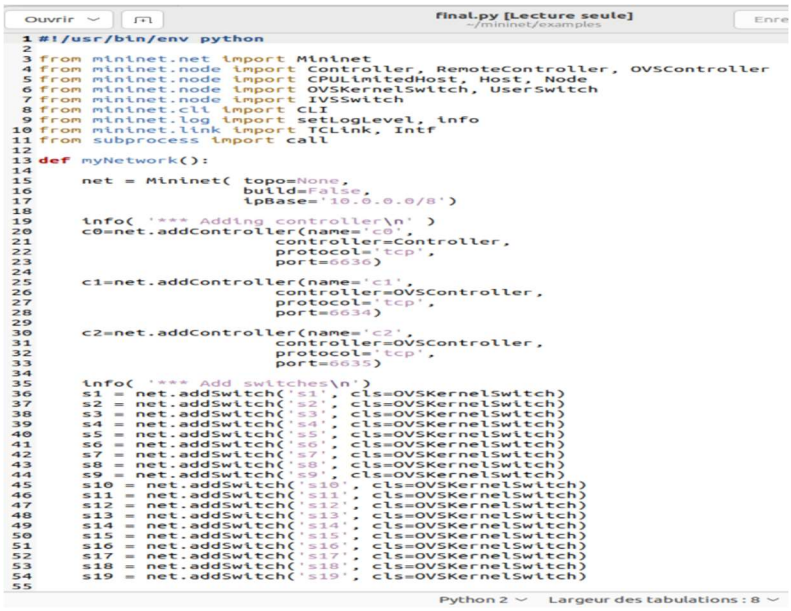


Figure 4.30: Script python

Figure 4.31: Création des switches avec python

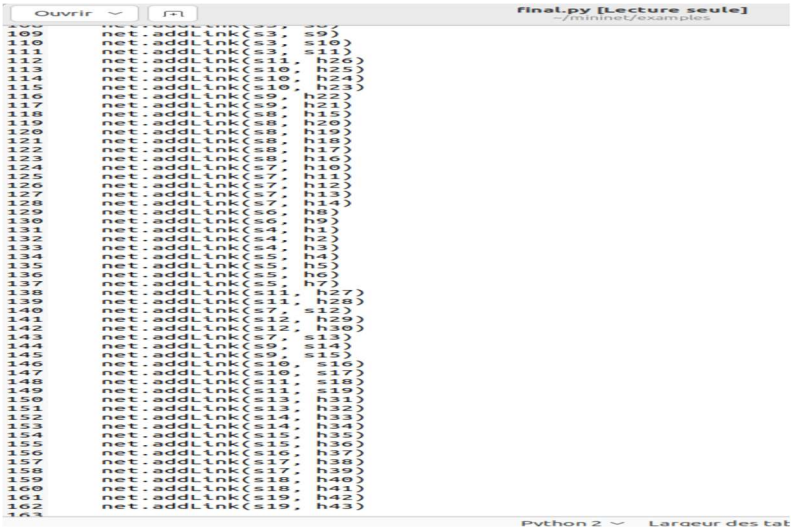


Figure 4.32: Les liaisons entre les périphériques

Figure 4.33: Starting and configuration

Nous avons trois contrôleurs. Dans cette expérience, nous utiliserons pour le contrôleur ‘c0’ ***OpenFlow référence*** par défaut intégré à Mininet et pour les contrôleurs ‘c1’ et ‘c2’ nous utilisons ***ovs contrôler*** Cependant, nous devons configurer chaque contrôleur afin qu'il utilise un port différent.

Cliquer avec le bouton droit sur chaque contrôleur et sélectionner *Propriétés* dans le menu qui s'affiche. Le numéro de port par défaut pour chaque contrôleur est 6633. Modifier-le pour que les numéros de port utilisés par les contrôleurs *c0*, *c1* et *c2* sont : 6636, 6634 et 6635.Comme les figures **(4.34) (4.35) (4.36) (4.37) (4.38)** illustrer.

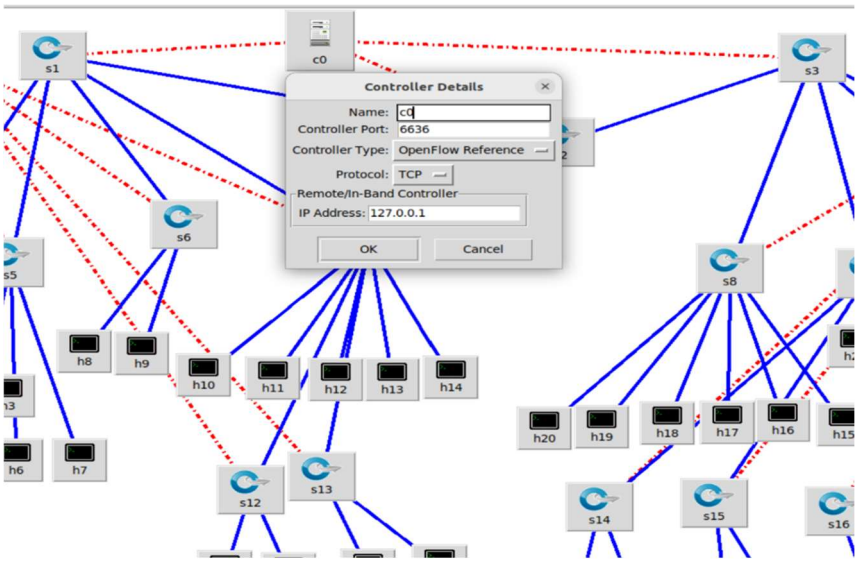


Figure 4.34: Configurer le contrôleur C0

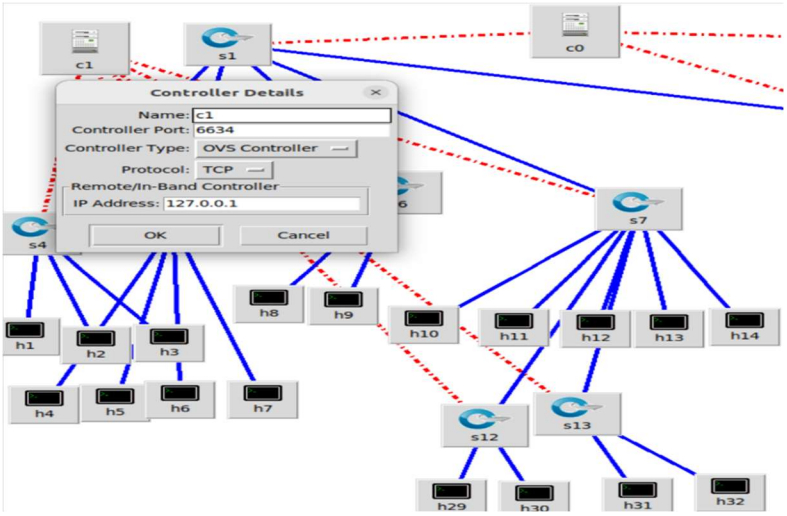


Figure 4.35: Configurer le contrôleur C1

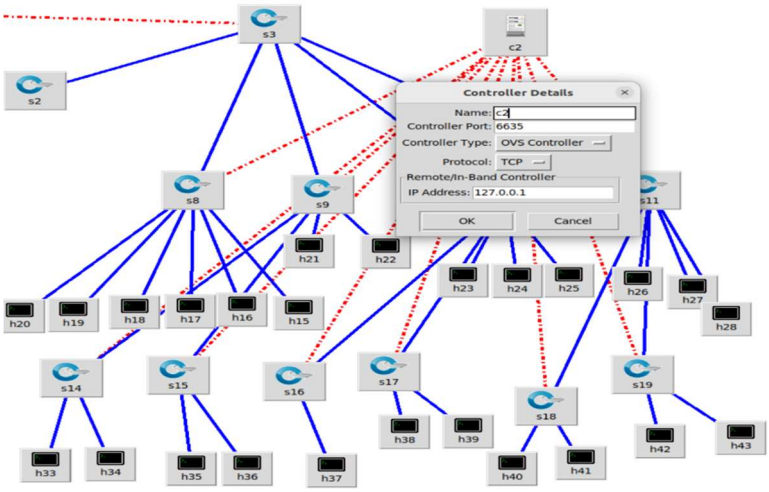


Figure 4.36: Configurer le contrôleur C2

Nous utilisons la commande de menu MiniEdit, *Édition → Préférences*. Dans la boîte de dialogue qui s'affiche, Pour définir les préférences de MiniEdit.

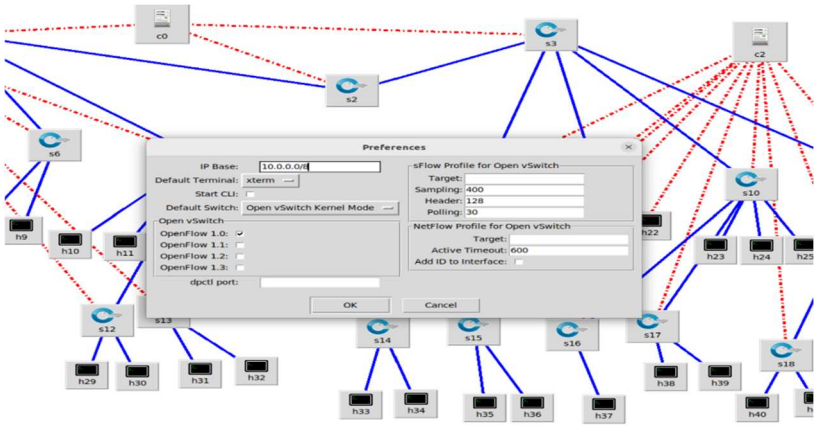


Figure 4.37: Préférences

Dans notre scénario, nous utiliserons la CLI et laisserons tous les autres paramètres aux valeurs par défaut.

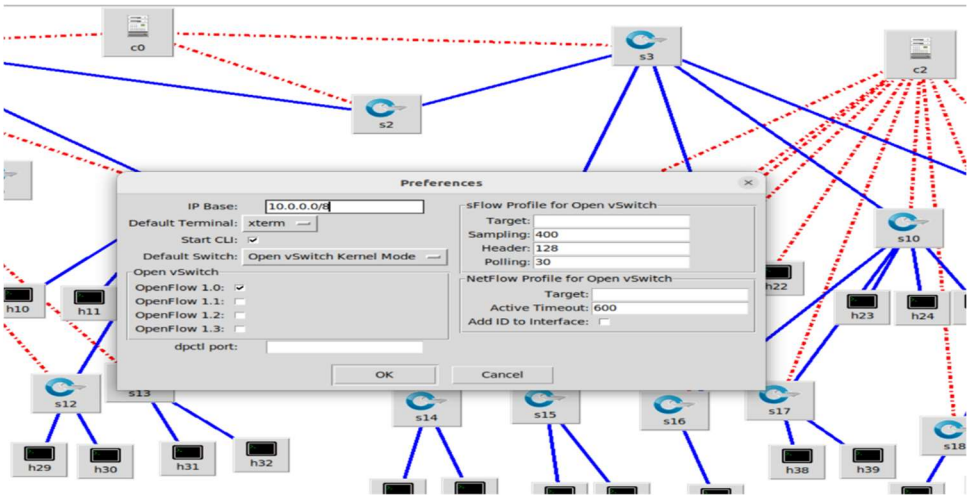


Figure 4.38: Les modifications sur préférences

Nous avons maintenant un scénario de réseau défini par logiciel qui devrait permettre à chaque hôte de communiquer avec n'importe quel autre hôte du réseau.

Nous allons enregistrer le fichier de topologie MiniEdit afin de pouvoir charger ce scénario dans MiniEdit à l'avenir.

Pour démarrer le scénario de simulation, cliquons sur le bouton *run* sur l'interface graphique de MiniEdit. Dans la fenêtre du terminal à partir de laquelle nous avons démarré MiniEdit, nous verrons des messages indiquant la progression du démarrage de la simulation, puis l'invite Miniedit CLI (car nous avons coché la case *Start CLI* dans la fenêtre des préférences de MiniEdit). Comme illustrer la figure **(4.39)**

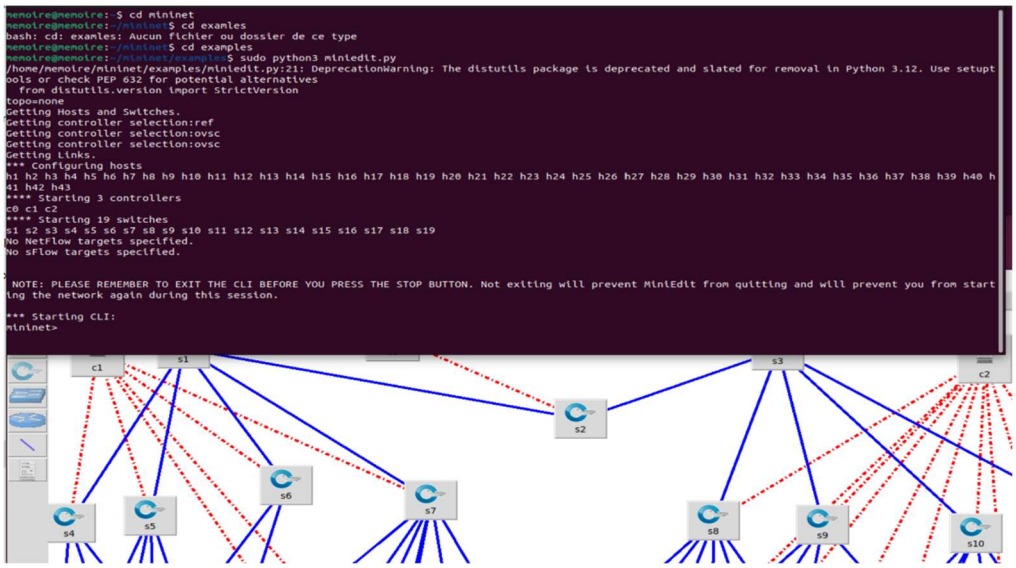


Figure 4.39: Exécution de la topologie

4.8.2 Les configuration open Vswitch

Pour vérifier que tous les commutateurs dans la simulation est configure correctement. Nous

pouvons exécuter la commande de menu MiniEdit. *Exécuter 'Run' → Afficher le résumé OVS 'show OVS summary'* pour voir une liste des configurations de commutateur. Comme la figure **(4.40)** illustrer.

Dans ce cas, nous pouvons vérifier que chaque commutateur écoute le bon contrôleur sur le bon port.

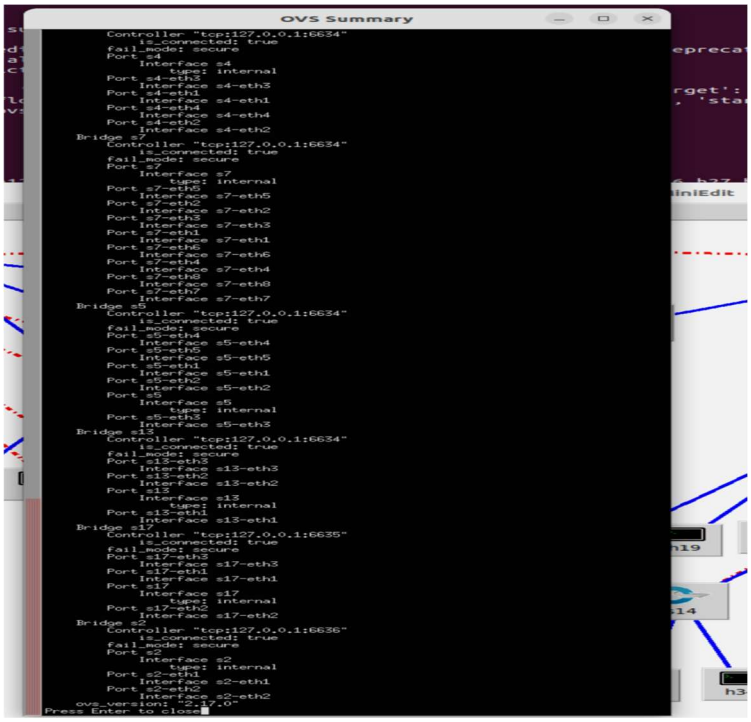


Figure 4.40: Résume OVS

4.8.3 Les tables de flux des commutateurs

Pour afficher les tables de flux de certaines des commutateurs en utilisant la commande mininet> dpctl dump-flows, illustre par la figure **(4.41)**

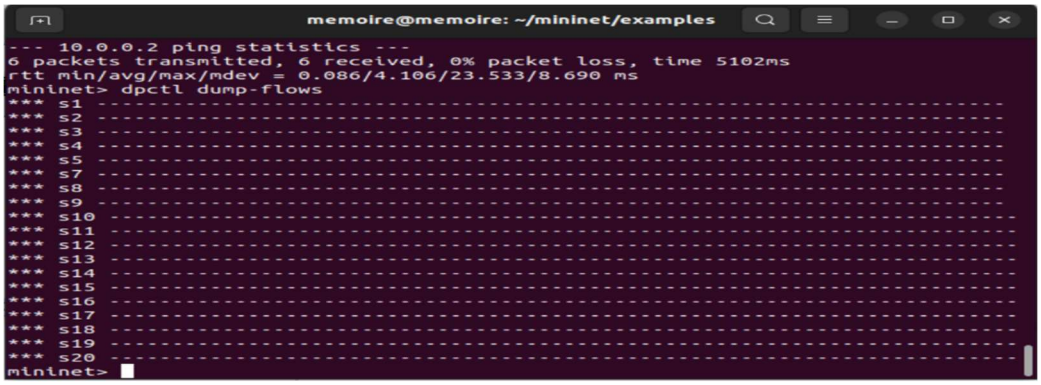


Figure 4.41: Les tables de flux

Nous vérifions le tableau de flux sur le commutateur s1 à l'aide des commandes ci-dessous. Il devrait être vide. Comme illustre la figure ***(4.42)***

>dpctl dump-flows s1

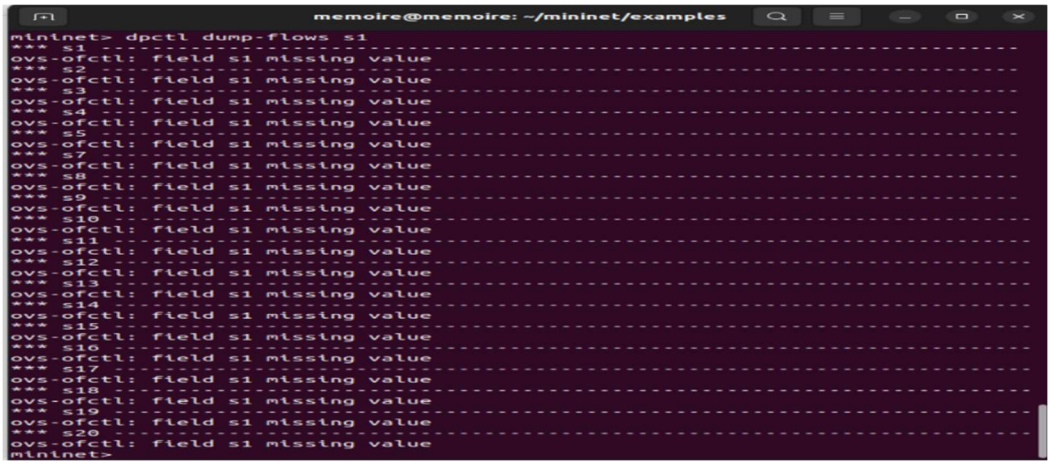


Figure 4.42:Table de flux S1

4.8.4 Exécuter des programmes pour générer et surveiller le trafic (Latance)

Avec un clic droit sur les hôtes *h1* et *h43* on ouvre une fenêtre xterm dans l'interface graphique de MiniEdit et sélectionnez *Terminal* dans le menu qui apparaît. Comme montre la figure **(4.43),** Dans la fenêtre *h1* xterm, démarrons un *Wireshark* avec la commande suivante,#wireshark

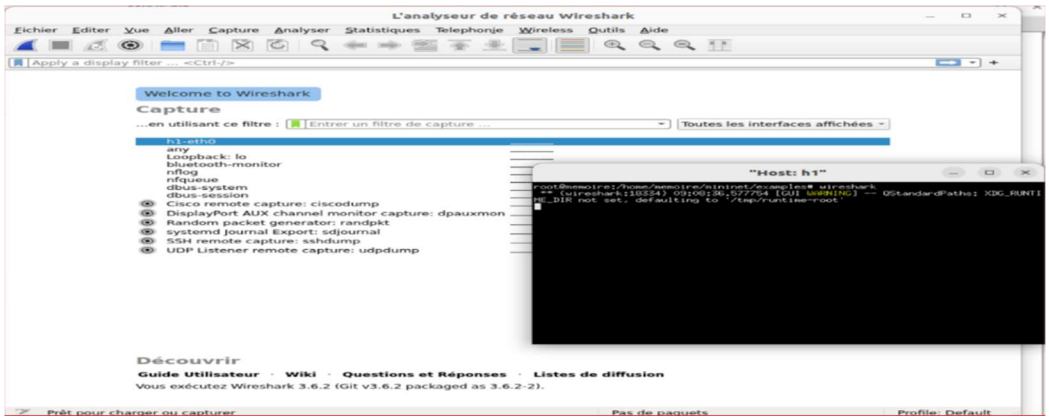


Figure 4.43:Wireshark

Dans la fenêtre *h8* xterm, démarrons une trace de paquet avec la commande suivante : #tcpdump, comme illustre la figure **(4.44)**

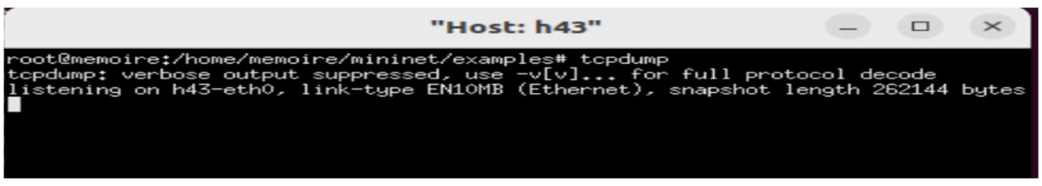


Figure 4.44: Tcpdump

Nous avons démontré deux méthodes différentes de surveillance du trafic sur les ports Ethernet virtuels de chaque hôte.

Ensuite, exécutons une commande *ping* pour envoyer le trafic entre les hôtes (h*1* et *h43), (h17 et h39) et (h8 et h27)* qui sont illustrés dans les figures ***(4.45) (4.46) (4.47) (4.48) (4.49) (4.50)***. Dans la fenêtre de la console MiniEdit, saisissez la commande suivante :

```
Mininet>h1 ping h43
```

```
mininet> h1 ping h43
PING 10.0.0.43 (10.0.0.43) 56(84) bytes of data.
64 bytes from 10.0.0.43: icmp_seq=1 ttl=64 time=13.1 ms
64 bytes from 10.0.0.43: icmp_seq=2 ttl=64 time=4.83 ms
64 bytes from 10.0.0.43: icmp_seq=3 ttl=64 time=0.571 ms
64 bytes from 10.0.0.43: icmp_seq=4 ttl=64 time=0.127 ms
64 bytes from 10.0.0.43: icmp_seq=5 ttl=64 time=0.116 ms
64 bytes from 10.0.0.43: icmp_seq=6 ttl=64 time=0.110 ms
64 bytes from 10.0.0.43: icmp_seq=7 ttl=64 time=0.093 ms
64 bytes from 10.0.0.43: icmp_seq=8 ttl=64 time=0.132 ms
64 bytes from 10.0.0.43: icmp_seq=9 ttl=64 time=0.155 ms
64 bytes from 10.0.0.43: icmp_seq=10 ttl=64 time=0.139 ms
64 bytes from 10.0.0.43: icmp_seq=11 ttl=64 time=0.153 ms
```

Figure 4.45:Test de connectivite entre h1 et h43

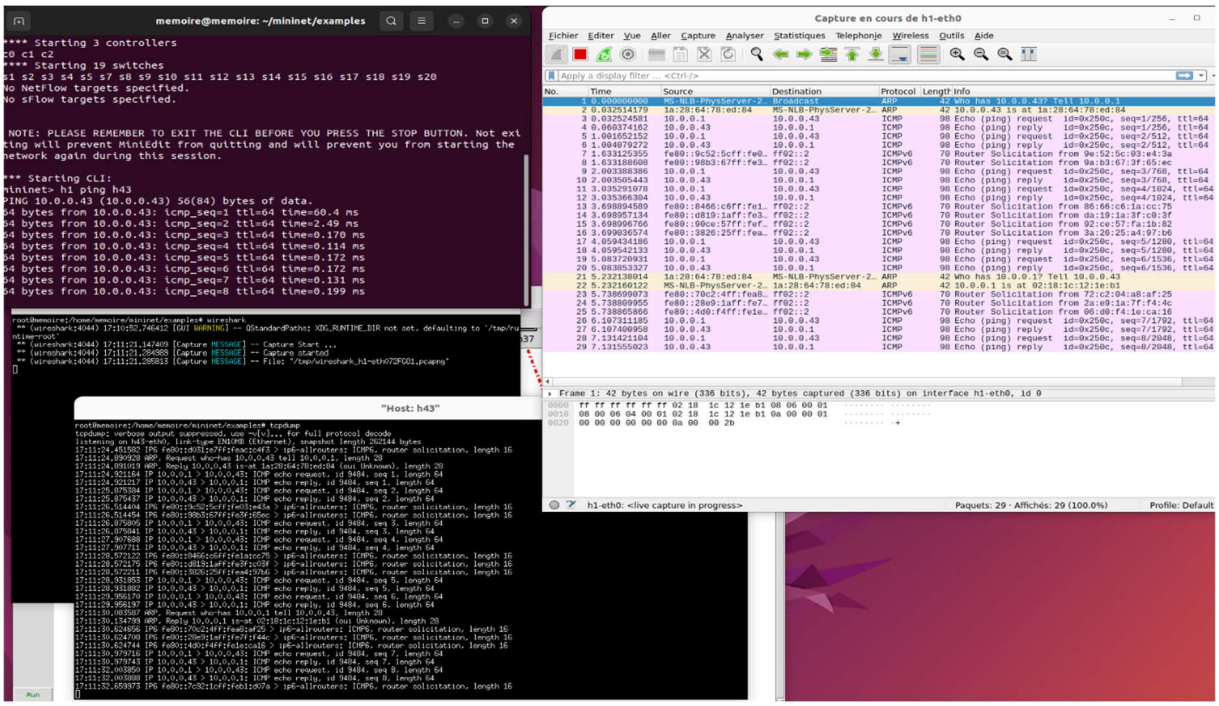


Figure 4.46:Surveillance du trafic généré par la commande ping entre h1 et h43

```
Mininet> h17 ping h39
```

```
mininet> h17 ping h39
PING 10.0.0.39 (10.0.0.39) 56(84) bytes of data.
64 bytes from 10.0.0.39: icmp_seq=1 ttl=64 time=11.3 ms
64 bytes from 10.0.0.39: icmp_seq=2 ttl=64 time=0.969 ms
64 bytes from 10.0.0.39: icmp_seq=3 ttl=64 time=0.092 ms
64 bytes from 10.0.0.39: icmp_seq=4 ttl=64 time=0.124 ms
64 bytes from 10.0.0.39: icmp_seq=5 ttl=64 time=0.073 ms
64 bytes from 10.0.0.39: icmp_seq=6 ttl=64 time=0.157 ms
64 bytes from 10.0.0.39: icmp_seq=7 ttl=64 time=0.066 ms
64 bytes from 10.0.0.39: icmp_seq=8 ttl=64 time=0.121 ms
64 bytes from 10.0.0.39: icmp_seq=9 ttl=64 time=0.122 ms
64 bytes from 10.0.0.39: icmp_seq=10 ttl=64 time=0.256 ms
64 bytes from 10.0.0.39: icmp_seq=11 ttl=64 time=0.300 ms
64 bytes from 10.0.0.39: icmp_seq=12 ttl=64 time=0.065 ms
64 bytes from 10.0.0.39: icmp_seq=13 ttl=64 time=0.116 ms
64 bytes from 10.0.0.39: icmp_seq=14 ttl=64 time=0.091 ms
64 bytes from 10.0.0.39: icmp_seq=15 ttl=64 time=0.101 ms
64 bytes from 10.0.0.39: icmp_seq=16 ttl=64 time=0.088 ms
64 bytes from 10.0.0.39: icmp_seq=17 ttl=64 time=0.074 ms
64 bytes from 10.0.0.39: icmp_seq=18 ttl=64 time=0.077 ms
64 bytes from 10.0.0.39: icmp_seq=19 ttl=64 time=0.348 ms
64 bytes from 10.0.0.39: icmp_seq=20 ttl=64 time=0.081 ms
64 bytes from 10.0.0.39: icmp_seq=21 ttl=64 time=0.065 ms
64 bytes from 10.0.0.39: icmp_seq=22 ttl=64 time=0.181 ms
64 bytes from 10.0.0.39: icmp_seq=23 ttl=64 time=0.064 ms
64 bytes from 10.0.0.39: icmp_seq=24 ttl=64 time=0.060 ms
64 bytes from 10.0.0.39: icmp_seq=25 ttl=64 time=0.071 ms
```

Figure 4.47:Test de connectivite entre h17 et h39

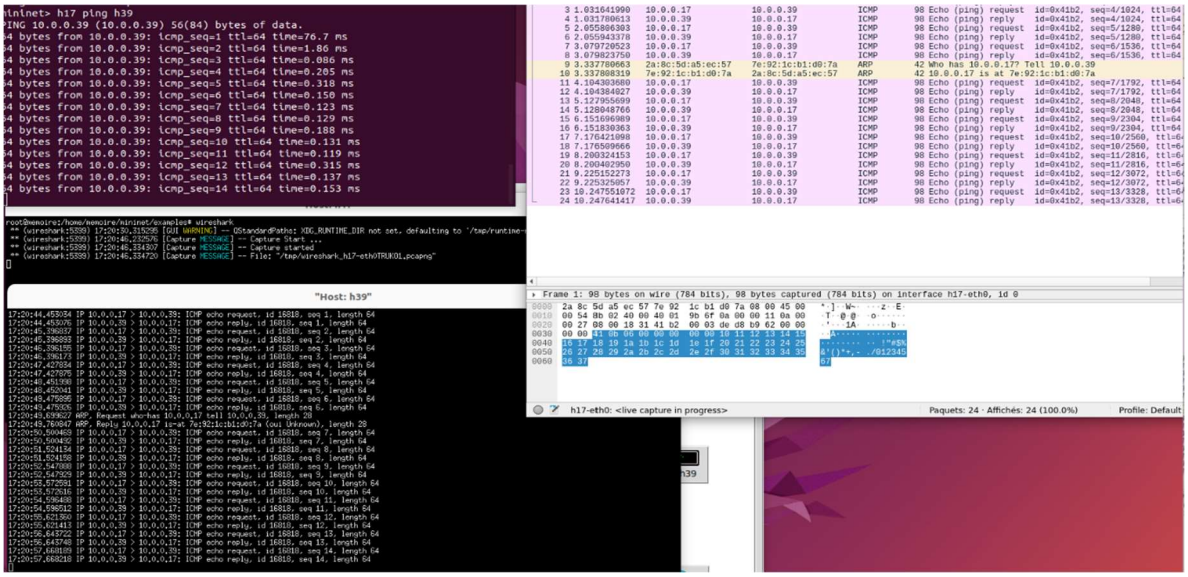


Figure 4.48:Surveillance du trafic généré par la commande ping entre h17 et h39

Mininet> h8 ping h27

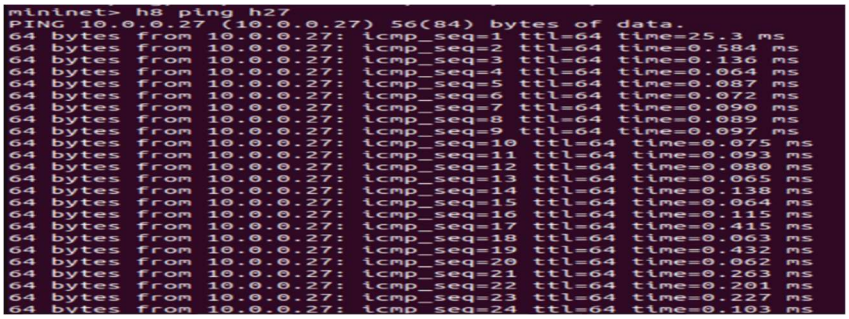


Figure 4.49:Test de connectivite entre h8 et h27

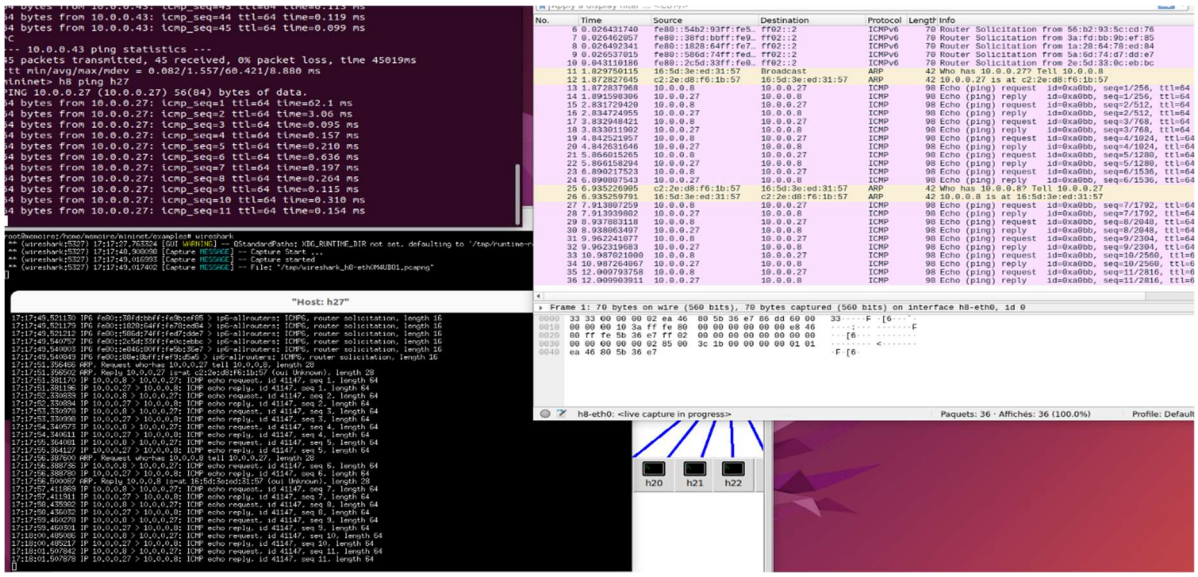


Figure 4.50:Surveillance du trafic généré par la commande ping entre h8 et h27

4.8.5 **Test d'accessibilité**

Le réseau se comporte comme prévu en termes d'accessibilité, ce qui confirme la fiabilité des restrictions appliquées. Illustre par les figures **(4.51) (4.52) (4.53)**.

Figure 4.51:Test d'accessibilité 1

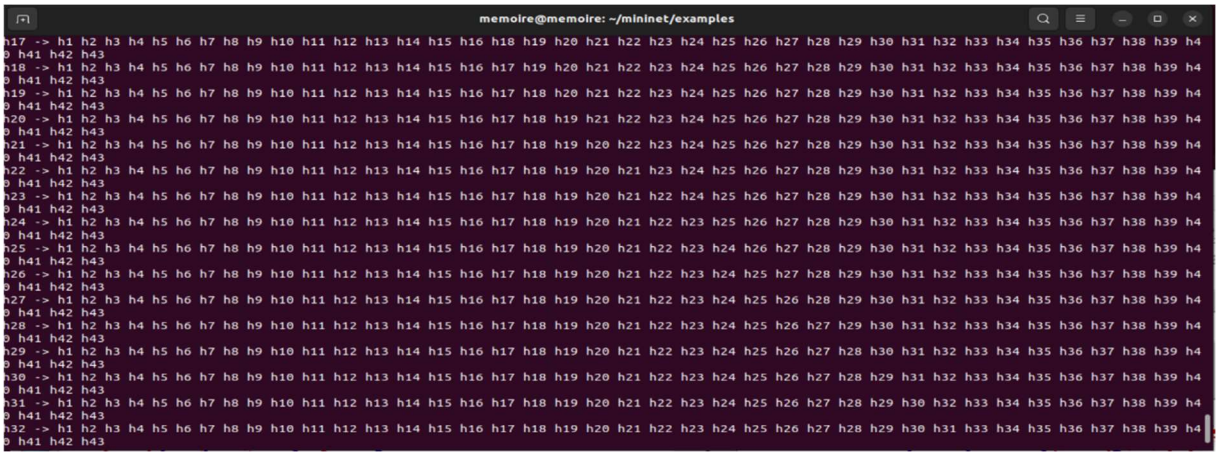


Figure 4.52:Test d'accessibilité 2

Figure 4.53:Test d'accessibilité 3

4.8.6 Test de la résilience (redondance)

Il est également possible de couper les liens entre nos machines, très pratique pour tester la résilience d'une application distribuée. Avec la commande suivante : mininet>Link

Comme illustre la figure **(4.54)**

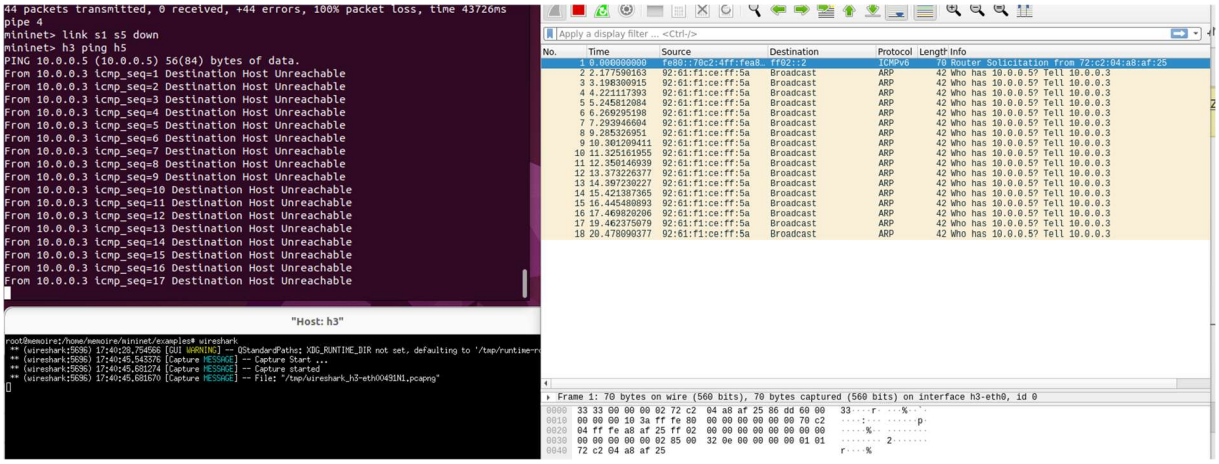


Figure 4.54:Link down

Et pour relier entre s1 et s5, en utilisant la commande mininet>Link s1 s5 up qu'est présente dans la figure **(4.55)**

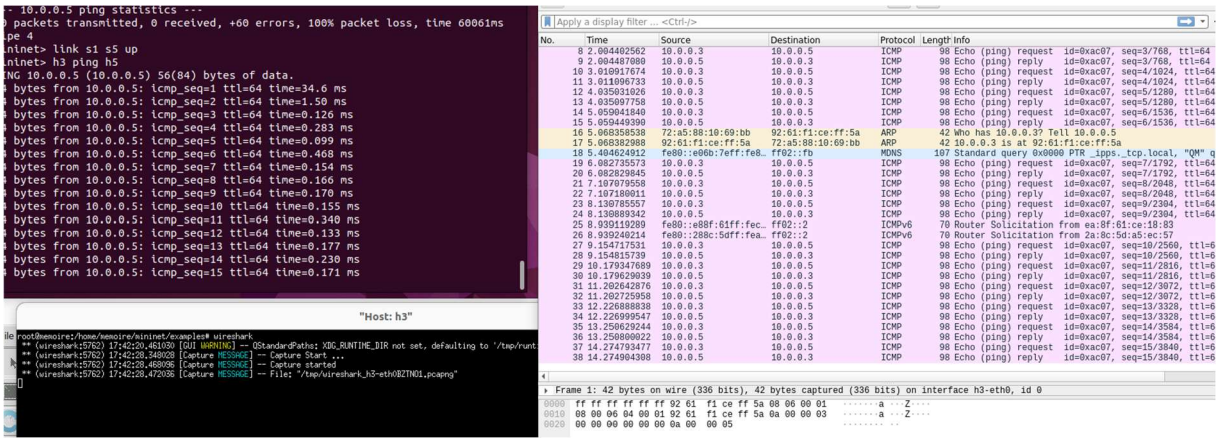


Figure 4.55:Link up

4.9 Installer le contrôleur RYU

Avant d'installer RYU nous installons *'pip'*, comme illustre la figure ***(4.56)*** et en utilisant la commande suivante :>sudo pip install –upgrade pip

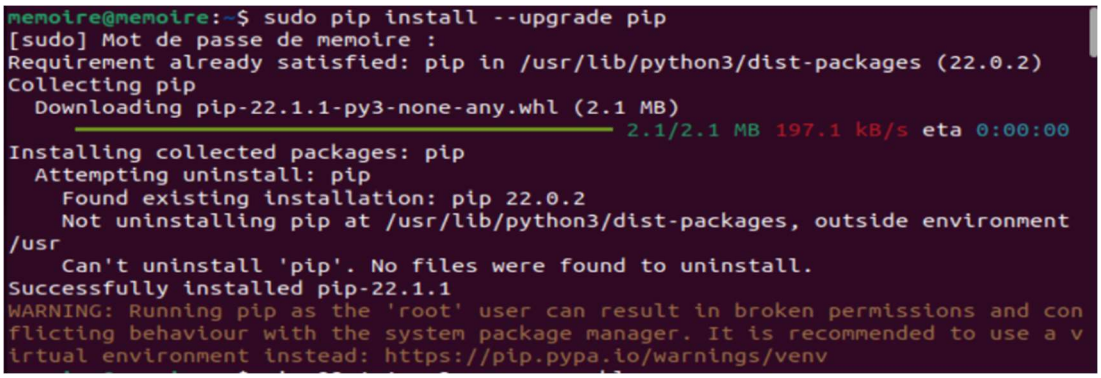


Figure 4.56:Installation du PIP

L'installation de Ryu est assez simple, nous utilisons la commande suivante :>pip install ryu

comme illustre la figure **(4.57)**

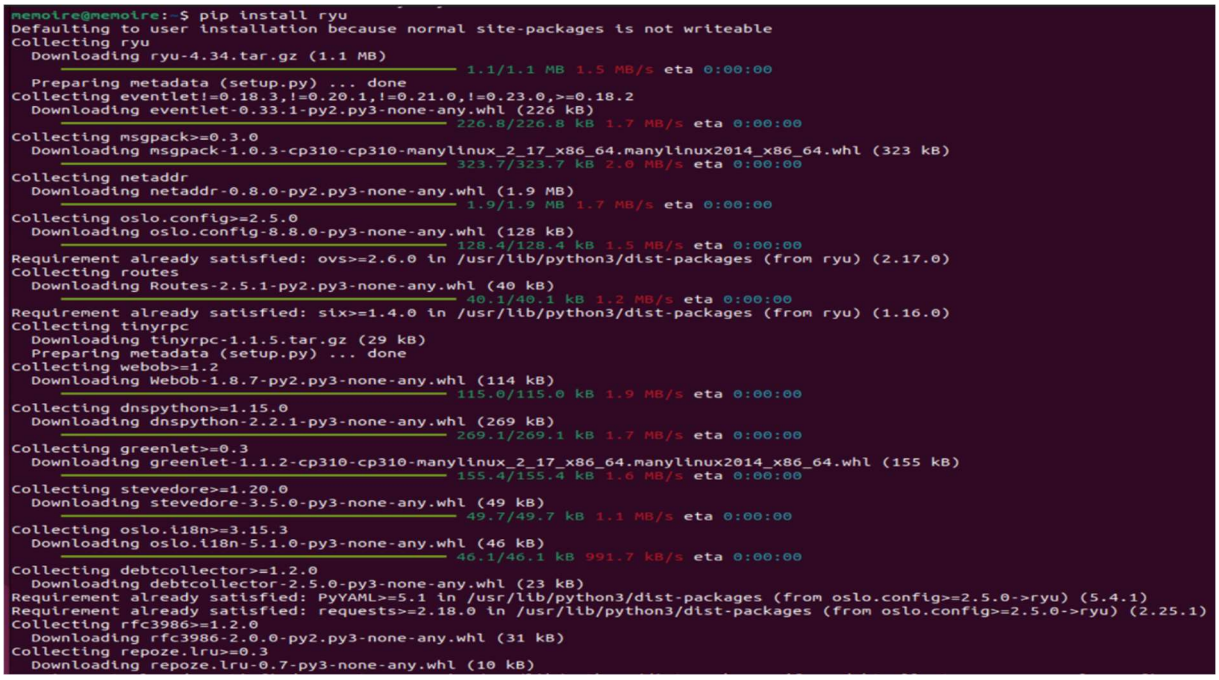


Figure 4.57:Installation du RYU

La figure **(4.58)** montre comment installer le controleur RYU depuis le code source

>sudo git clone https://github.com/faucetsdn/ryu.git

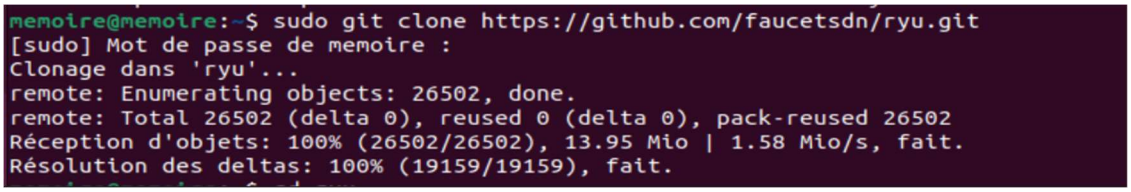


Figure 4.58:Installation du contrôleur RYU depuis le code source

Nous avons utilis cette commande pour ouvrire le ficher RYU dans le terminal :

>cd ryu

Puis nous avons insatall PIP dans RYU nous avons utilis la commande suivante :>sudo pip install -r tools/pip-requires comme montre la figure **(4.59)**

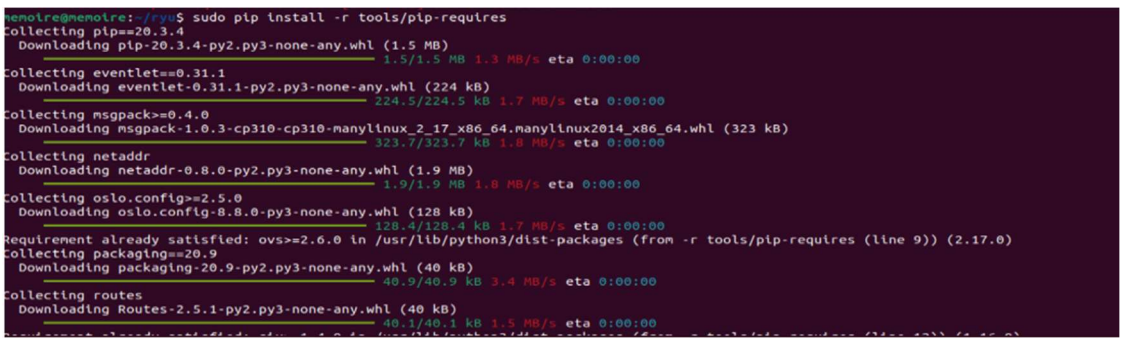


Figure 4.59:Installation de PIP dans le fichier RYU

Nous installons python 3 dans le fichier RYU, comme la figure **(4.60)** illustre et en utilisant

la commande suivante :>sudo python3 setup.py install

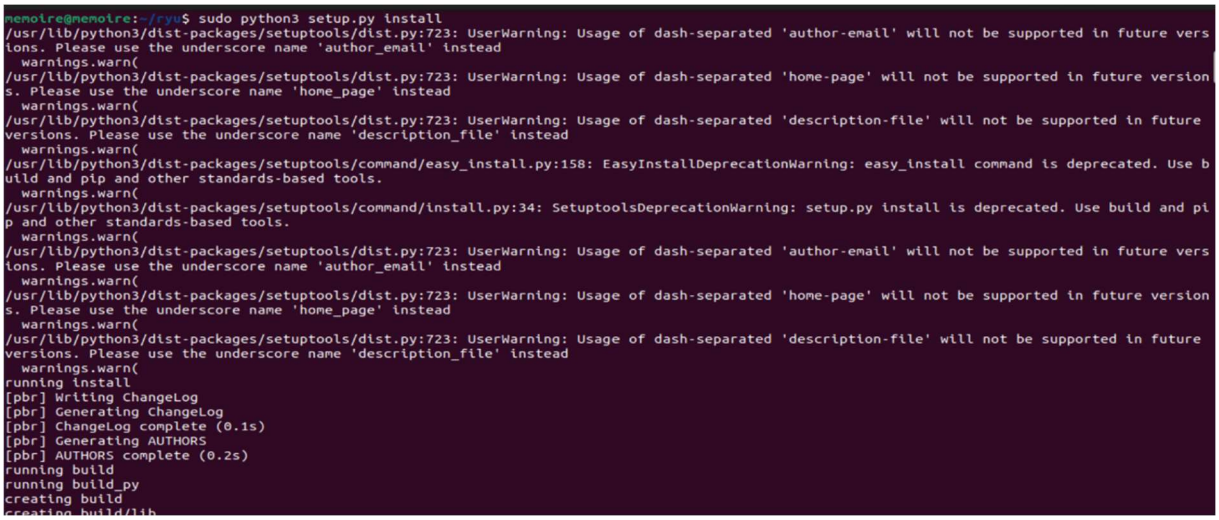


Figure 4.60:Installation du python 3

Pour écrire notre application RYU, nous tapons la commande suivante :

>ryu-manager

4.9.1 Exécuter des programmes pour générer et surveiller le trafic (Latence)

Dans les figures **(4.61) (4.62) (4.63) (4.64) (4.65) (4.66)**, nous utilisons les mêmes commandes que celles du contrôleur Pox.

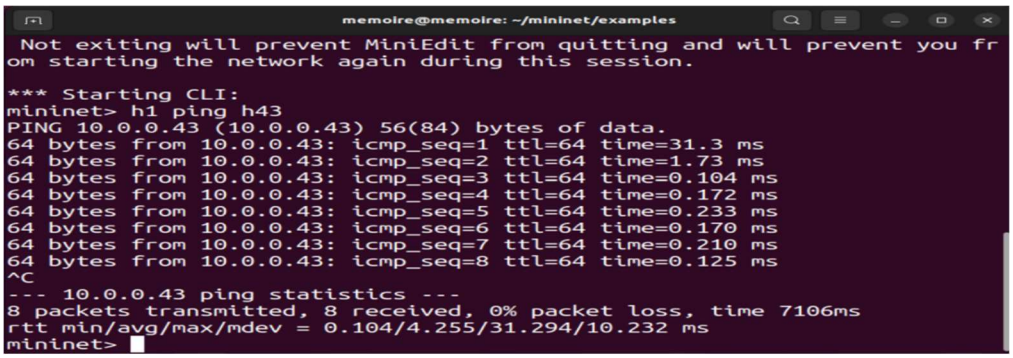


Figure 4.61:Test la connectivité entre h1 et h43 pour le contrôleur RYU

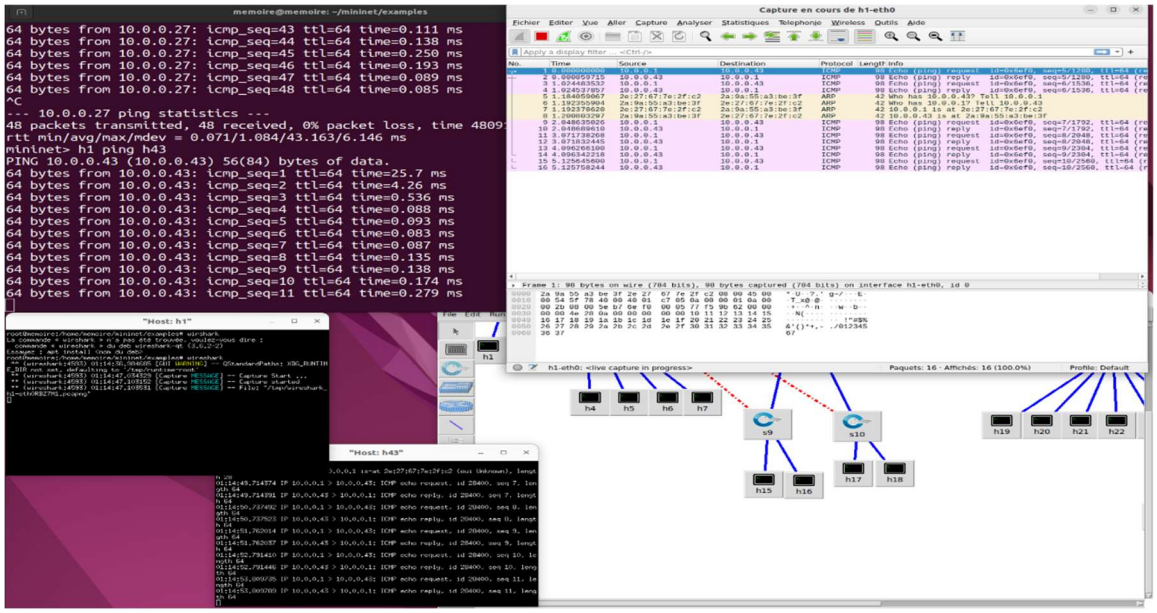


Figure 4.62:Surveillance du trafic généré par la commande ping entre h1 et h43 pour le contrôleur RYU

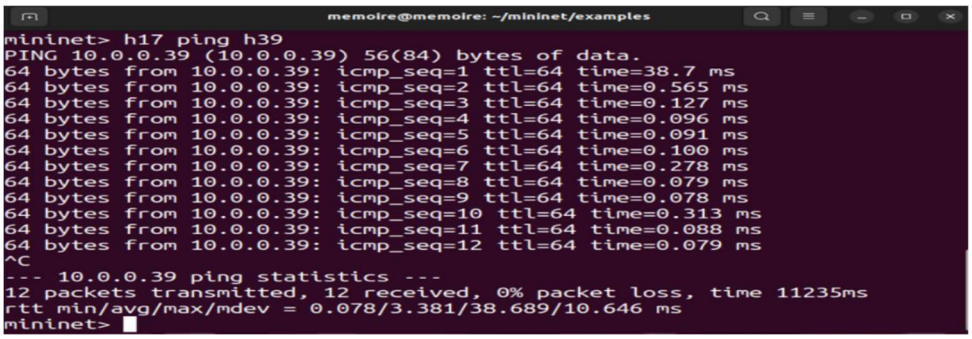


Figure 4.63:Test de connective entre h17 et h39 pour le contrôleur RYU

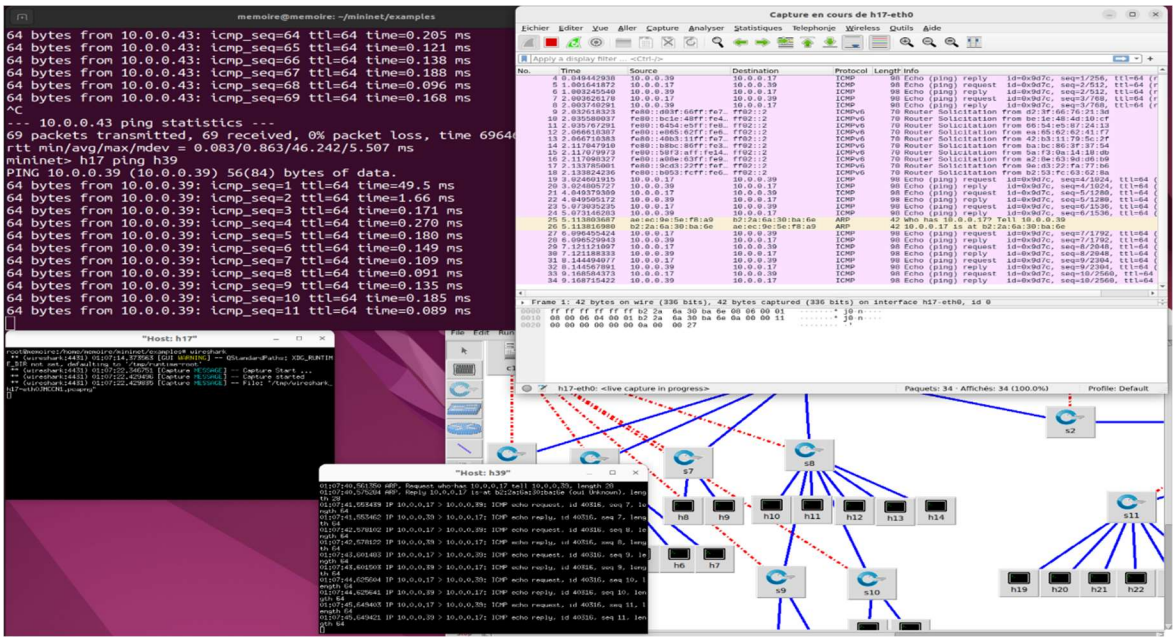


Figure 4.64:Surveillance du trafic généré par la commande ping entre h17 et h39 pour le contrôleur RYU

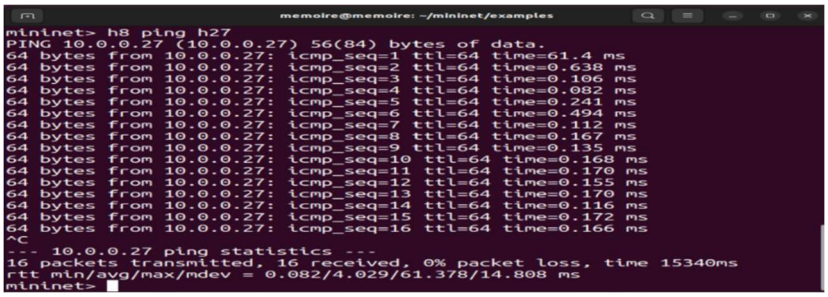


Figure 4.65:Test de connectivite entre h8 et h27 pour le contrôleur RYU

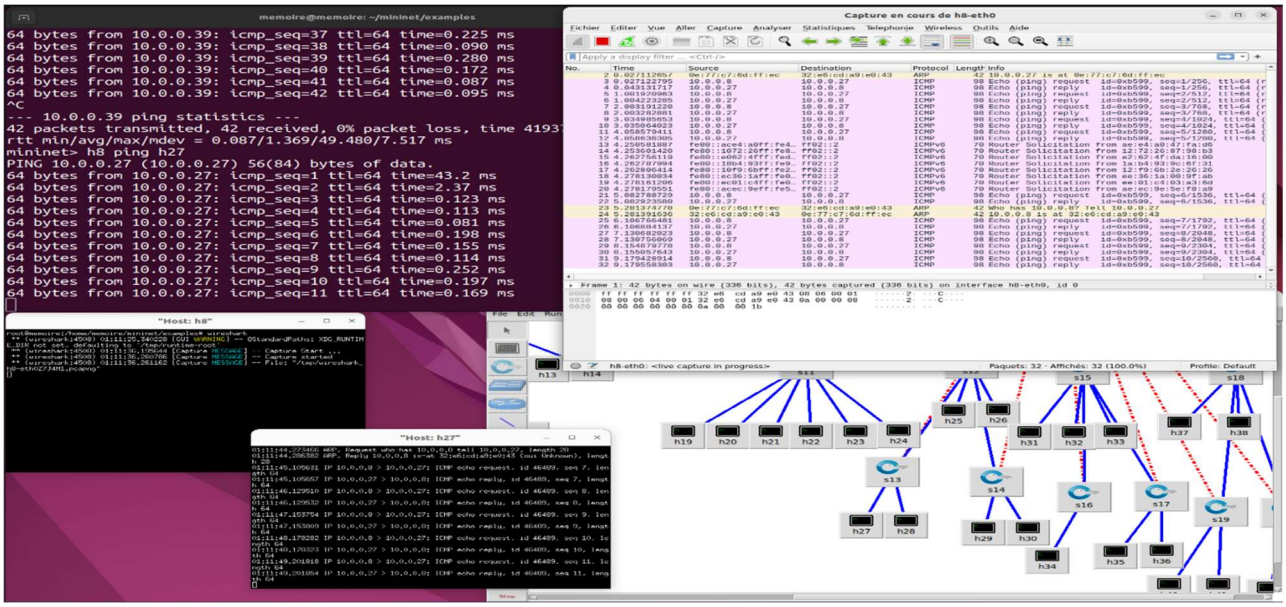


Figure 4.66:Surveillance du trafic généré par la commande ping entre h8 et h27 pour le contrôleur RYU

Nous avons remarqué que le temps de connectivite entre les deux hôtes dans le contrôleur RYU est lev par-rapport au temp de connectivite du contrôleur POX.

4.9.2 Test d'accessibilité

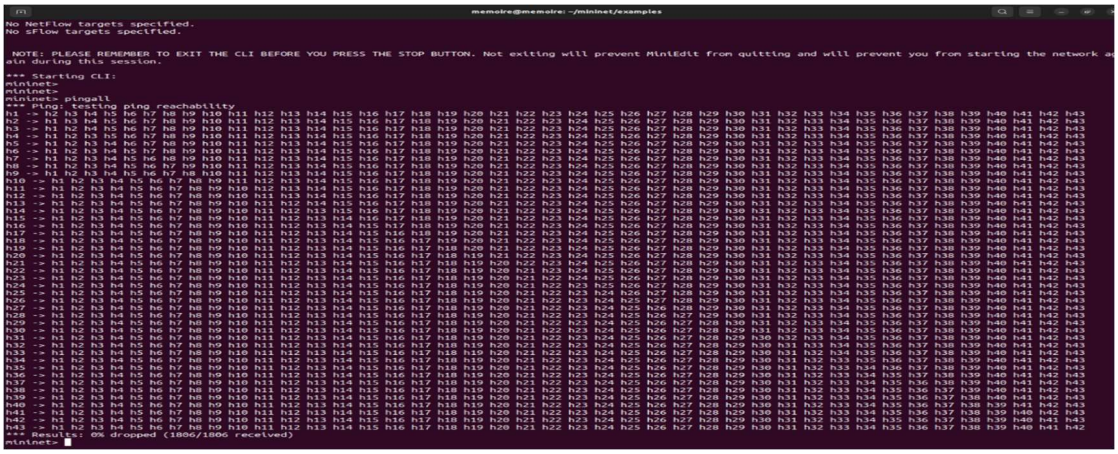


Figure 4.67:Test d'accessibilité pour le contrôleur RYU

D'âpre la figure **(4.67)** en remarquant que les contrôleurs POX et RYU ont la même

accessibilité.

4.9.3 Test de la résilience (redondance)

Dans les figures **(4.68) (4.69)** nous avons présenté la redondance de contrôleur RYU.

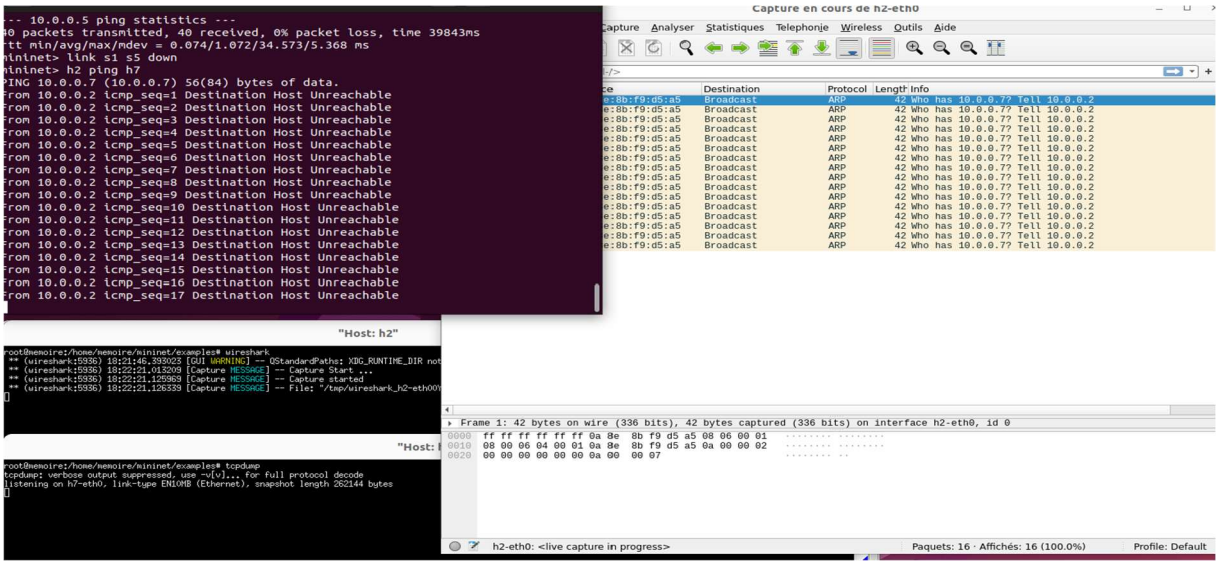


Figure 4.68:Link down dans le contrôleur RYU

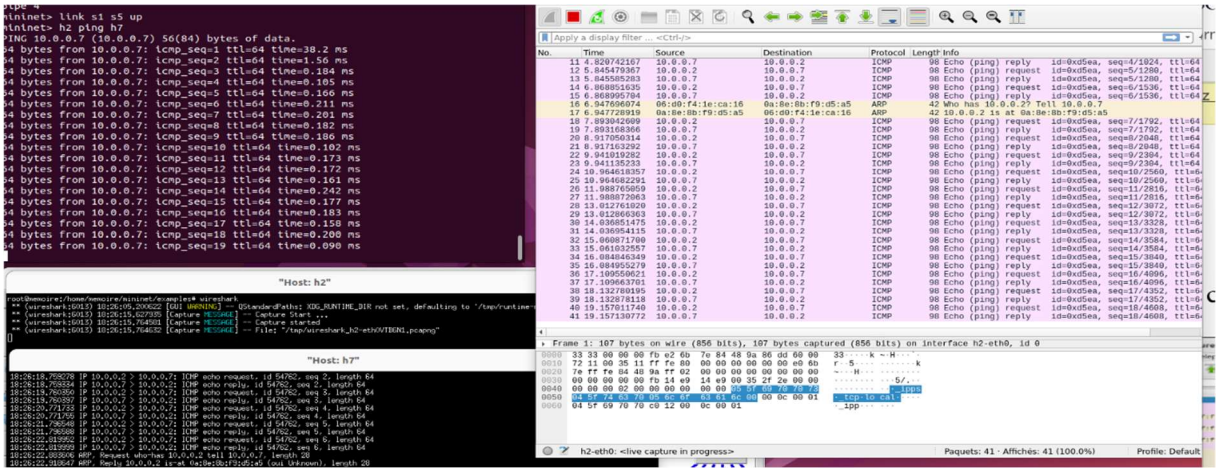


Figure 4.69:Link up dans le contrôleur RYU

Les contrôleurs POX et RYU ont la même redondance.

4.10 Discussion

L'analyse comparatif des performances des contrôleurs est une tâche difficile dans ce travail, d'objectif d'administrer un réseau de type SDN. Nous avons comparé qualitativement et quantitativement deux contrôleurs avec Mininet et *WireShark*. Il a été possible de comparer la latence, l'accessibilité et la redondance, entre deux open source protocole tel que RYU et Pox.

Les résultats montrent clairement que le contrôleur Ryu présente une latence élevée par rapport au contrôleur Pox, une faible latence et une réponse inadéquate à petit réseau avec de nombreux commutateurs, sa mise en œuvre n'est donc pas beaucoup recommandée. Alors nous

avons choisi le contrôleur Pox.

4.11 Conclusion

Dans ce chapitre nous avons utilisé différents équipements pour implémenter un réseau SDN. Ainsi nous avons fait une analyse comparative des performances des contrôleurs Ruy et Pox, d'où nous constatons que RYU n'est pas beaucoup recommandé pour cela nous choisissons le contrôleur Pox.

Dans la suite de notre mémoire nous allons réaliser un système de sécurité firewall dans un environnement SDN, en utilisant le contrôleur Pox.

Chapitre 5

Réalisation de Firewall dans un Environnement SDN

Chapitre 5 Réalisation de Firewall dans un environnement SDN

5 Introduction

La sécurité est un problème majeur dans les réseaux. Ce qui conduit les entreprises et les fournisseurs de services de dépenser beaucoup d'argent dans des firewalls très coûteux pour mettre en œuvre la sécurité.

Le réseau défini par logiciel (SDN) est une nouvelle structure qui peut faire économiser de l'argent aux entreprises et aux fournisseurs de services, réduire le temps de provisionnement de quelques jours à quelques minutes, fournir une gestion centralisée, permettre la programmabilité dans le réseau implicite.

Les firewalls peuvent être facilement mis en œuvre à l'aide des règles OpenFlow par défaut, dans cette deuxième partie de notre travail, nous réalisons un firewall qui est un moyen de mettre en œuvre le SDN. Nous avons utilisé le protocole OpenFlow dans un SDN-Firewall pour filtrer le trafic entre les hôtes selon des critères spécifiés, puis l'autoriser ou non à passer en utilisant le contrôleur POX fourni avec Mininet pour configurer nos politiques ou règles nécessaires et en utilisant les commutateurs pour filtrer le trafic entre les hôtes.

5.1 Firewall

Firewall est un système qui sécurise les paquets réseau entrants, qui proviennent de diverses sources, ainsi que les paquets réseau sortants. Il peut surveiller et contrôler le flux de donnée entrant dans le réseau à partir de différentes sources et fonctionne sur la base de règles prédéfinies.

Le Firewall est une sorte de mécanisme de cyber sécurité utilisé sur un réseau pour filtrer le trafic. Les Firewall sont utilisés pour différencier les nœuds du réseau des sources de trafic externes, des sources de trafic internes et même des applications spécifiques.

Les Firewall peuvent être logiciels, matériels ou basés sur le cloud, et chacun a ses propres avantages et inconvénients. L'objectif principal du pare-feu est d'empêcher les requêtes de trafic et les paquets de données malveillants tout en permettant le trafic à traverser ***(Ijiert, 2021).***

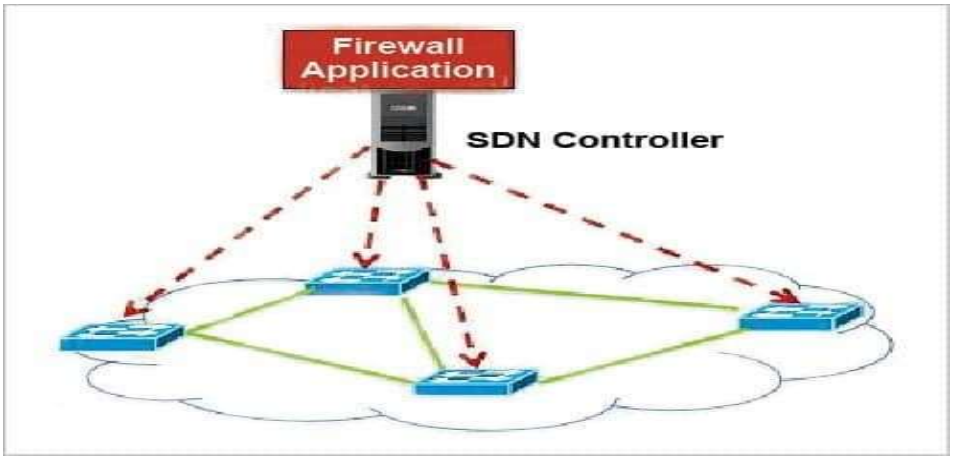


Figure 5.1: Architecture de base SDN-firewall

5.1.1 Type Firewall

Il existe plusieurs types de Firewall différents en fonction de leur fonction :

a) Filtrage des paquets

Selon des critères définis par l'utilisateur, chaque paquet entrant ou sortant du réseau est inspecté et autorisé ou refusé.

b) Circuit-Gateway Firewall

Lorsqu'une connexion TCP ou UDP est établie, cette procédure implémente des mesures de sécurité. Même s'il est incroyablement économe en ressources, le paquet lui-même n'est pas testé par ces pares-feux.

Ainsi, si un paquet transportait des logiciels malveillants, mais avait la bonne poignée de main pour TCP, il passerait directement par le Firewall c'est l'inconvénient majeur de ce Firewall.

c) Stateful Inspection Firewall

Pour établir un niveau de sécurité supérieur à celui des Firewall à filtrage de paquets ou Firewall de passerelle de circuit, ces pares-feux intègrent à la fois des technologies d'inspection de paquets et une poignée de main TCP authentification.

d) Firewall proxy

Dans la couche application, les Firewall proxy s'exécutent pour filtrer les données entrantes entre le réseau et la source du trafic. Ils sont distribués par une solution basée sur le cloud ou un autre système proxy.

Le Firewall proxy crée d'abord une connexion à la source du trafic et inspecte le paquet de données entrant, plutôt que d'avoir un lien de trafic directement.

5.2 Rôle SDN-Firewall

Nous identifions sept caractéristiques vitales qui un Firewall doit prendre en charge pour tirerparti des capacités réseau avancées activées par SDN.

a) Application centralisée des politiques

La fonctionnalité de base de tout Firewall est d'appliquer les politiques de sécurité dans un réseau. Ainsi, une SDN utilisant un matériel de Firewall traditionnel comme nœud dans le réseau n'exploite pas les fonctions avancées permis par SDN : centralisation et haute programmabilité. Les politiques spécifiées au niveau central peuvent aller d'une granularité fine à une grande grossièreté. Pour renforcer la sécurité d'un réseau, le Firewall convertit ces politiques en règles de flux qui sont ensuite installées sur le réseau.

b) Suivi de flux centralisé (contrôle de flux de bout en bout)

Les règles installées dans le réseau dirigent les paquets appartenant à un flux particulier de la source à la destination. Ainsi, tout chemin pris par les paquets appartenant à un flux, peut contenir plusieurs règles de flux provenant d'un ou plusieurs commutateurs. Pour efficacement appliquer la politique, le Firewall basé sur SDN garde une trace d'espace de chemin d'écoulement. Simplement confirmer la politique sur l'entrée et le port de sortie d'un commutateur ne peut pas garantir un conflit réussi détection, sans parler de la résolution.

En outre, la création d'un flux espace de chemin en tirant parti des garanties de centralisation pour prendre en compte les modifications qu'OpenFlow autorise dans l'en-tête champs d'un flux de réseau en route.

c) Résolution des conflits

Lors de la détection d'une violation dans le lors de l'ajout d'une nouvelle politique ou d'une règle de flux, un Firewall complet doit également fournir une résolution de conflit. Différentes instances, Flow Guard et SE-Floodlight, prouvent que le simplefait d'autoriser ou de refuser une mise à jour réseau n'est pas une conception de Firewall efficace. S'il existe un chemin dans le réseau sans entité en conflit, le Firewall doit effectuer dynamiquement la résolution pour donner un chemin mis à jour.

De même, la nouvelle mise à jour ne peut entraîner que partiellement une violation. Dans ce cas, le Firewall doit disséquer la mise à jour en ensembles autorisés/refusés et traiter l'ensemble autorisé. Donc, un Firewall complet basé sur SDN doit prendre en charge résolutions de conflits.

d) Gestion automatique des priorités

Différentes sources (par exemple, contrôleurs SDN, applications et administrateur) mettent à jour les politiques de sécurité dans un réseau. Attribuer une autorisation le niveau de chaque source est important pour éviter une faible priorité mise à jour de remplacement des politiques réseaux cruciaux. En outre, un Firewall intelligent devrait traiter ces autorisations automatiquement, les problèmes et offre une expérience utilisateur robuste.

e) Prise en charge de plusieurs locataires

Un réseau complexe, tel que des centres de données, peut contenir plusieurs sous-réseaux et plusieurs locataires pour différents services. Gérer plusieurs locataires n'est pas une pré-occupation particulière dans un réseau traditionnel aussi différent des Firewall sont dédiés à chaque réseau locataire. Cependant, dans SDN, le contrôleur a une vue centralisée de l'ensemble du réseau topologie et le pare-feu se charge de centraliser l'application des politiques de sécurité. Le Firewall basé sur SDN devrait être capable de générer une distinction entre les sous-réseaux même si leurs plages d'adresses se chevauchent.

f) Évolutivité et simultanéité

Dans un environnement dynamique et évolutif réseau, les mises à jour des politiques de sécurité ne sont pas toujours séquentielles. Plusieurs threads utilisateur ou applications s'exécutant à l'intérieur d'un contrôleur SDN effectue des mises à jour simultanées pour modifier le Firewall politique ou politique de flux. En cas de mises à jour contradictoires, un manque de la gestion de la concurrence pourrait conduire à des règles de faible priorité en cours de traitement avant de chevaucher des règles de priorité plus élevée. Le problème avec les mises à jour simultanées devient un problème lorsque les applications de Firewall sont déployées dans les réseaux d'entreprise, où il y a plusieurs pipelines qui accèdent et mettent à jour les mêmes magasins de données de configuration.

g) Support dynamique

Le maintien des états des connexions actives donne un avantage certain à la fiabilité d'un Firewall. Les états de connexion peuvent être obtenus à l'aide des informations de Communication OpenFlow. Les informations sur les états de connexion doivent être formulées dans des règles de flux limitées dans le temps pour un réseau fournissant une granularité plus fine dans les règles de Firewall et un mécanisme de détectionde

violation plus précis ***(Arizona, 2018).***

5.3 Implémentation et réalisation du Firewall

L'objective de notre deuxième partie pratique est pour tester nos règles de Firewall, nous avons créé deux scénarios de topologie en script python qui inclut tous les composants du réseau y compris le contrôleur POX à distance. Dans nos lignes d'importation de modules, nous devons d'abord importer Remonte Controller à la place du contrôleur standard, puis le transmettre à la classe Mininet. Ajout d'adresses MAC aux hôtes pendant l'initialisation nous permet de créer rapidement des règles de couche 2 pour notre Firewall.

Nous exécutons un script python avec POX contrôleur sous environnement MININET pourréaliser un Firewall.

5.3.1 Mettre en place une topologie à l'aide du Mininet

a) Scénario 1 : single topologie

La topologie que nous allons implémenter est une typologie simple avec un contrôleur, un switch et 10 hosts nous avons exécuté la ligne de commande suivante : $ sudo mn --topo single,10 --controller=remote, comme montre la **figure (5.2)**.

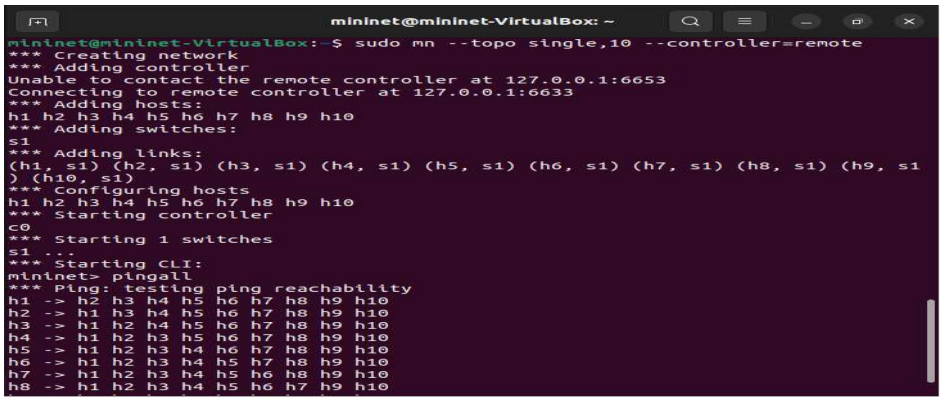


Figure 5.2 : Single topologie

Signal topologie est illustré ci-dessous dans la **figure (5.3)**

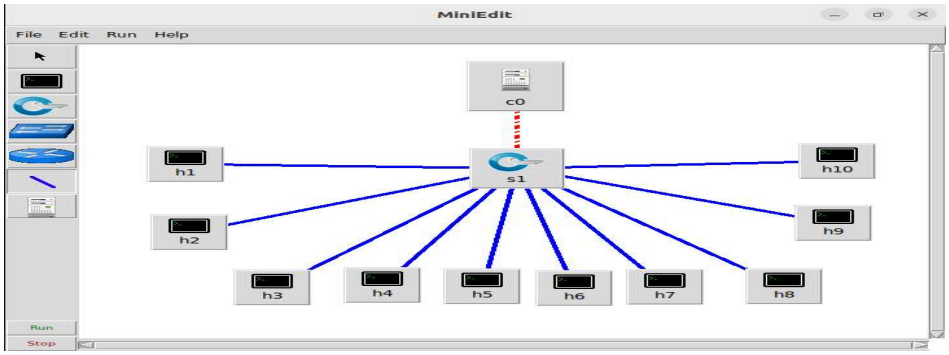


Figure 5.3 : Single topologie sur miniedit

Nous avons enregistré le code python de notre single topologie sous le nom 'single.py' et nous avons modifié le contrôleur Mininet par défaut par un contrôleur à distance Pox, les figures (5.4), (5.5) et (5.6) illustrent le code python après la modification de contrôleurs et l'ajout d'adresseMAC.

Figure 5.4 : Code single topologie après modification du contrôleur

Puis nous avons ajouté des adresses MAC simples lors de l'initialisation des hôtes afin que nous puissions facilement les configurer pendant la création de règles pour notre Firewall de couche 2.

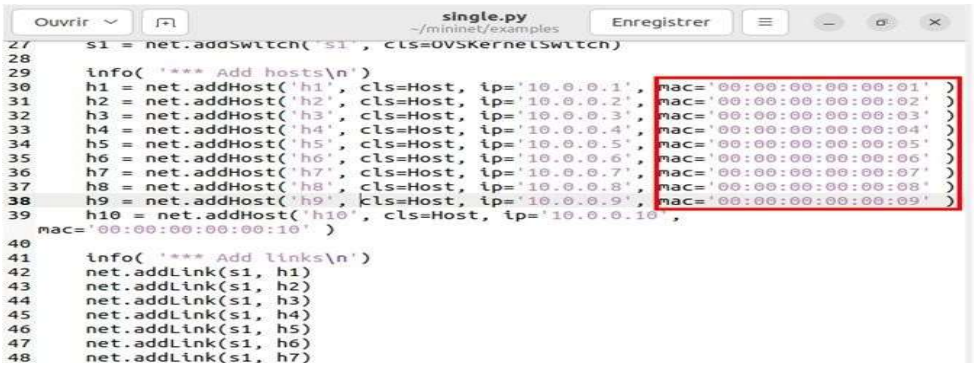


Figure 5.5 : L'ajout des adresse MAC au code python

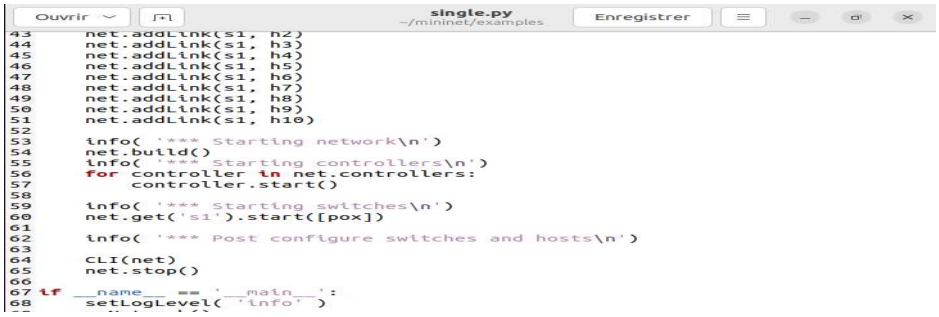


Figure 5.6 : Suite code python single.py

Nous exécutons un ping avant l'ajout des règles Firewall, pour vérifier que la communicationentre les hôtes fonctionne, comme illustré dans la **figure (5.7)**.

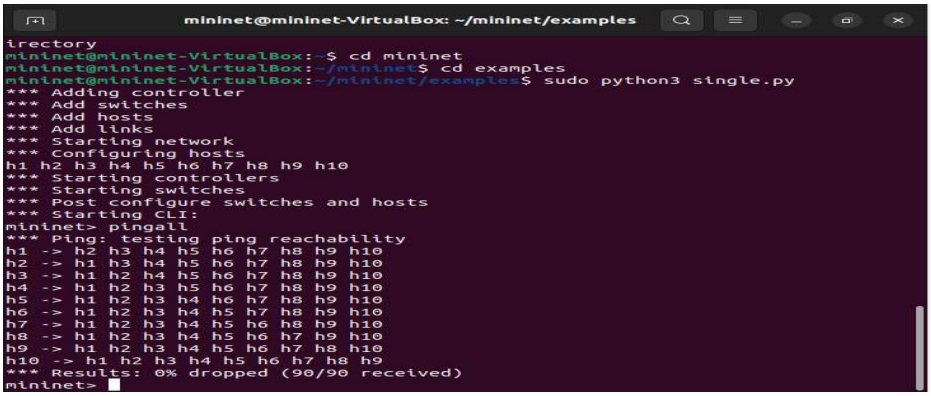


Figure 5.7 : Ping avant l'ajout des règles firewall

5.3.1.1 Règles de Firewall et mise en œuvre de POX

Pour que nous puissions facilement définir des règles de couche 2 pour notre Firewall, nous avons écrit le code python illustré ci-dessous **figure (5.8)** :

Ensuit nous avons décidé quelles règles à appliquer via le Firewall. Il convient

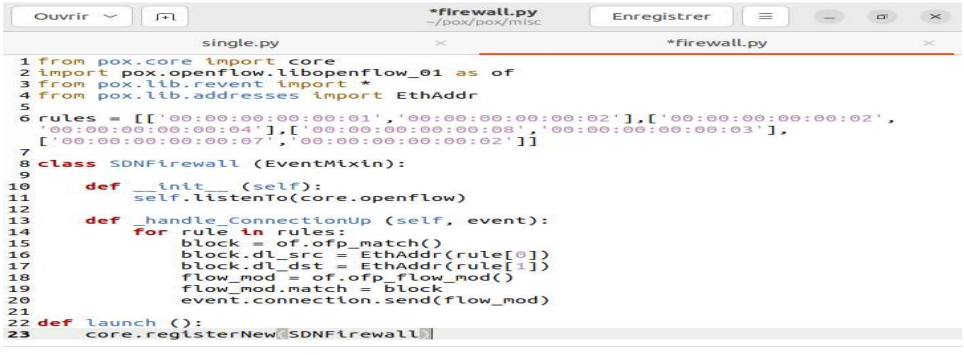


Figure 5.8 : Code python de notre firewall

également de noter qu'il s'agirait d'un blocage bidirectionnel, et si une règle existe, le contrôleur s'assure que le commutateur ne laisse pas le trafic entrant ou sortant de l'un ou l'autre hôte aller vers "l'autre" par le placement des adresses Mac chacune d'entre elles sous

forme de lignes dans une liste, afin de pouvoir parcourir chaque règle séparément.

Ainsi pour notre Firewall, nous pouvons avoir les 4 règles suivantes :

- h1 et h2 se bloquent
- h2 et h4 se bloquent
- h2 et h7 se bloquent
- h3 et h8 se bloquent

La figure (5.9) montre les règles Firewall ajoutées sur notre code python.

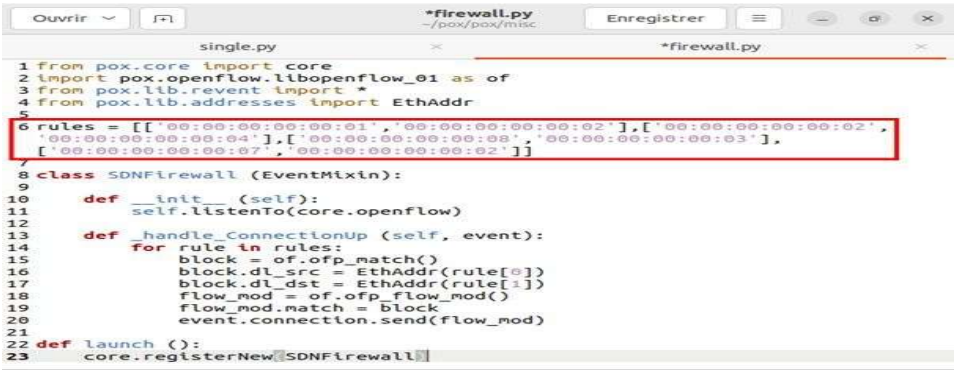


Figure 5.9 : Regle firewall applique

Le code python firewall a été enregistré dans le répertoire POX sous le nom ‘firewall.py’

La figure (5.10) : représente l’enregistrement de code Firewall dans le répertoire PO

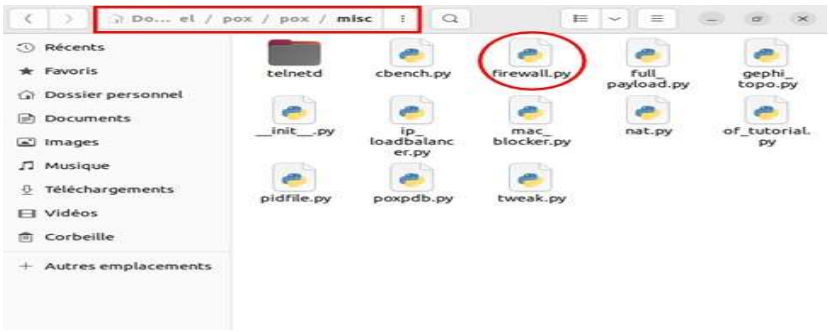


Figure 5.10 : Emplacement code firewall.py dans le répertoire POX

5.3.1.1.1 Exécution du SDN-Firewall

Tout d’abord nous démarrons l'émulation de topologie Mininet et le contrôleur POX puisimplémenter le Firewall en mode debug, comme suivant :

1. Nous avons Exécuté `sudo python single.py` dans le terminal pour initialiser les hôtes et les commutateurs et démarrer la CLI Mininet, comme représentés dans la **figure (5.11)**

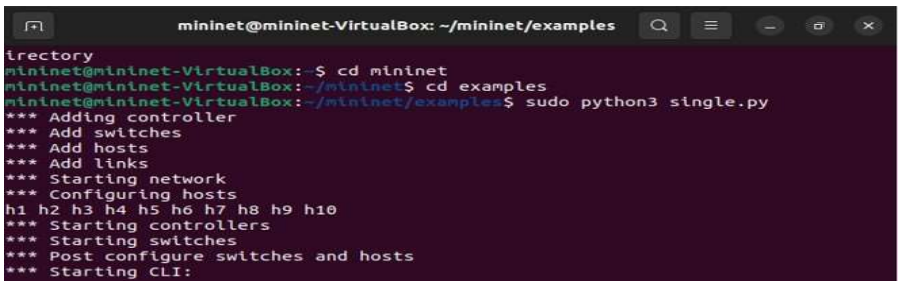


Figure 5.11 : Execution single.py

2. Nous exécutons le contrôleur dans le répertoire POX, pour implémenter notre Firewall par la commande suivante :

`. /pox.py log.level --DEBUG openflow.of_01 forwarding.l2_learning misc.firewall` comme représentée dans la **figure (5.12)**.

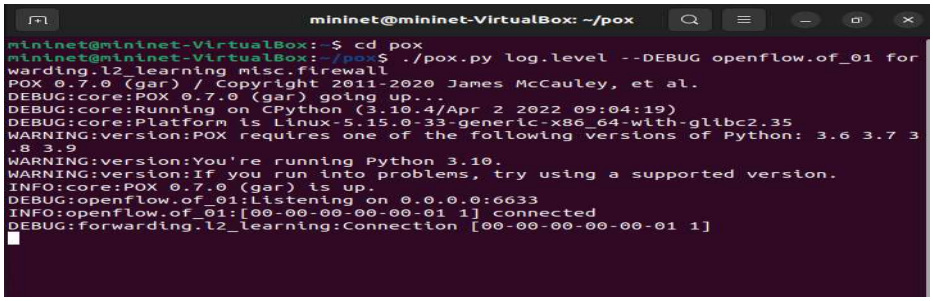


Figure 5.12 : Exécution de contrôleur

Méthode 1

Nous Revenons à la CLI Mininet et tapons `pingall` pour voir si notre Firewall fonctionne réellement qui est illustré dans la **figure (5.13)**.

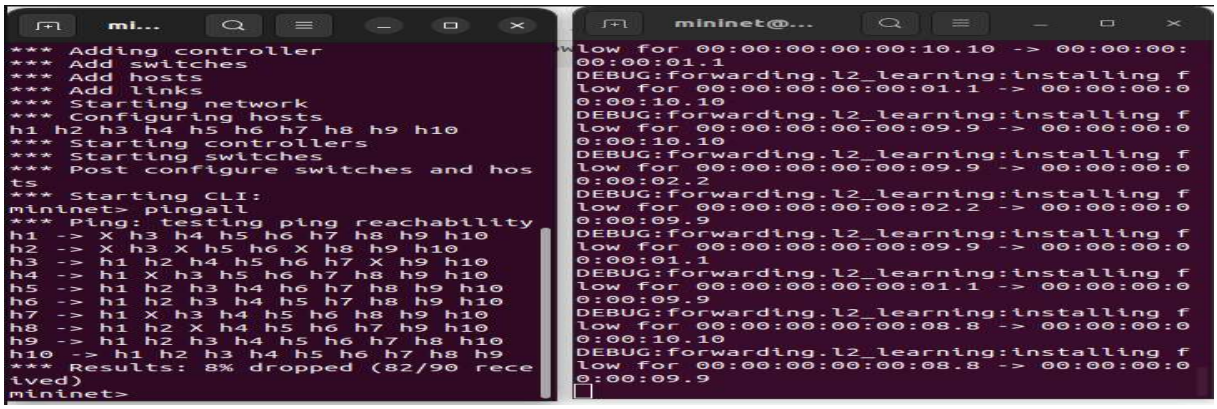


Figure 5.13 : Fonctionnent réel de firewall

Méthode 2

Nous faisons un ping entre chaque deux hosts, qui est illustré dans les **figures (5.14), (5.15), (5.16)**

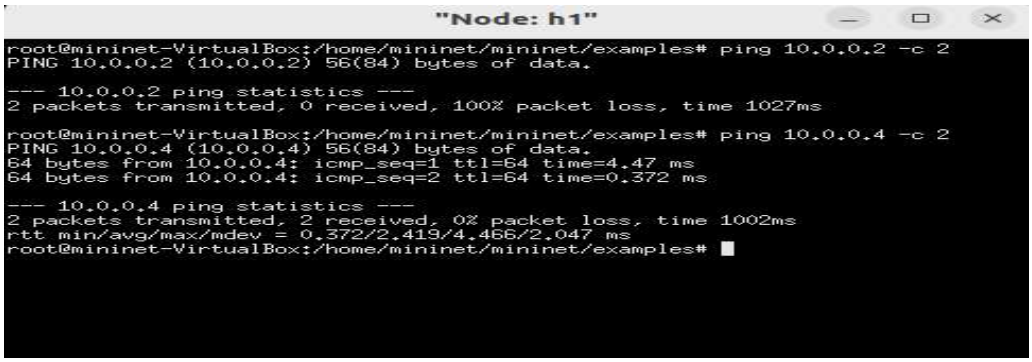


Figure 5.14 : Ping entre h1 et h2 et entre h1 vers h4

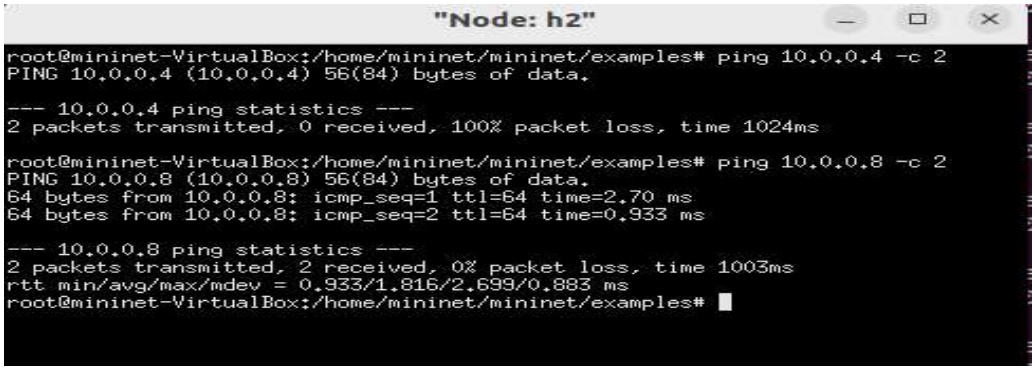


Figure 5.15: Ping entre h2 et h4 et entre h2 et h8

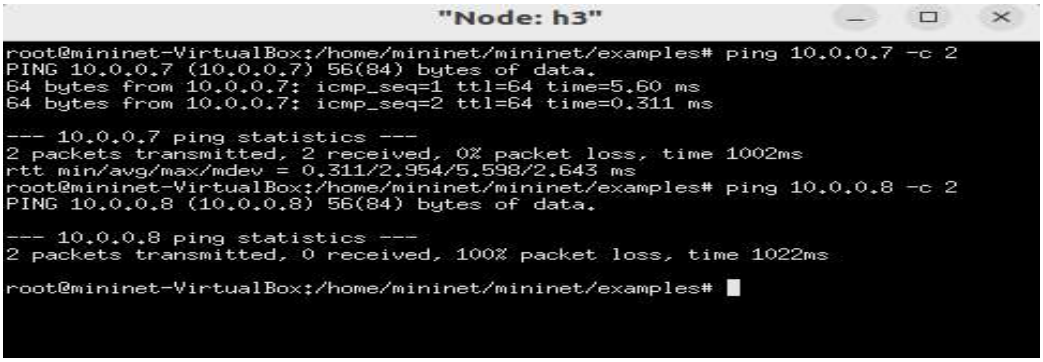


Figure 5.16 : Ping entre h3 et h7 et entre h3 et h8

Discutions

Selon les résultats des règles d'implémentation de notre Firewall, nous avons affirmé que h1et h2, h2 et h4, h3 et h8, h2 et h7 n'ont jamais pu se connecter, aussi nous voyons qu'il y a une perte de paquets de 8% acceptable. Donc nous avons réussi la mise en place de notre Firewall- SDN.

b) Scénario 2 : Topologie personnalisée

1. Dans ce scénario 2 nous avons suit les mêmes étapes que celles du scénario 1, mais avec une typologie personnalisée qui confirme la mise en œuvre de notre solution SDN avec trois contrôleurs pour gérer toute la charge. La charge est répartie entre les trois contrôleurs et avoirmoins de charge sur un seul contrôleur et 19 Switch OpenFlow dans Mininet et 43 hôtes. Les contrôleurs sont des contrôleurs POX qui aident à simuler le banc d'essai. La topologie illustrée dans la figure ci-dessus :

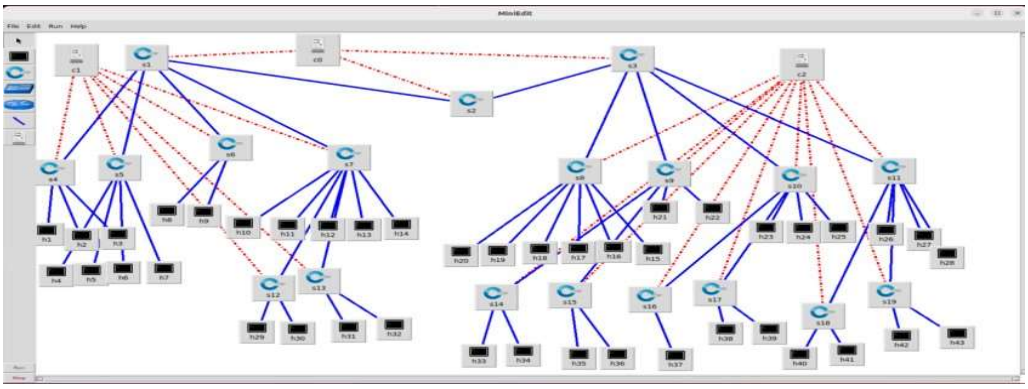


Figure 5.17: Topologie personnalisée

2. Nous avons ajouté des adresses MAC pour que nous puissions réaliser les règles de Firewall qui sont illustrées dans les **figures suivantes (5.18), (5.19), (5.20) et (5.21)**:

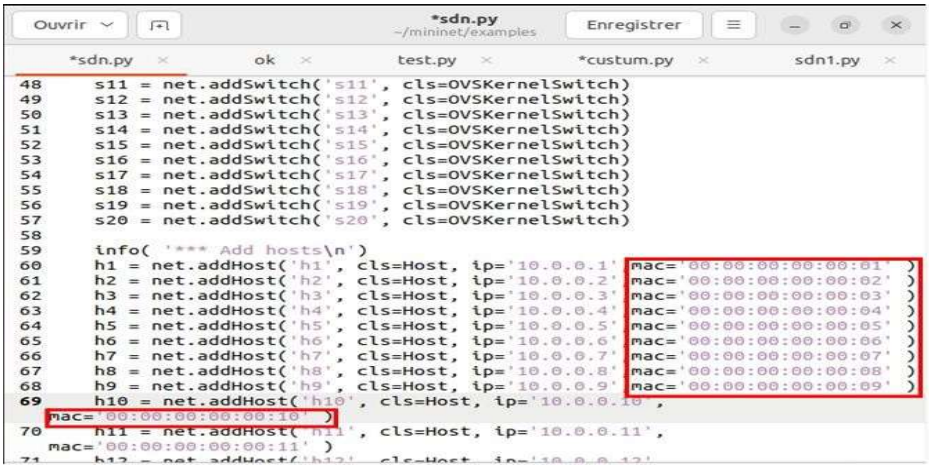


Figure 5.18 : Ajouter l'adresse MAC

Figure 5.19 : Ajouter d'adresse MAC suite 1

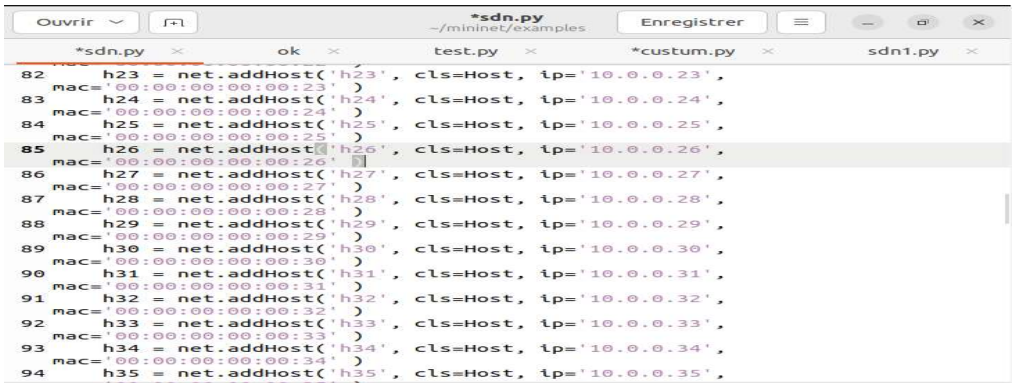


Figure 5.20 : Ajouter d'adresse MAC suite 2

Figure 5.21 : Ajout d'adresse MAC suite

Pour confirmer le déroulement correct de communication entre les hôtes, nous exécutons pingall avant l'ajout des règles Firewall comme illustré ci-dessous **figure (5.22) et figure (5.23**).

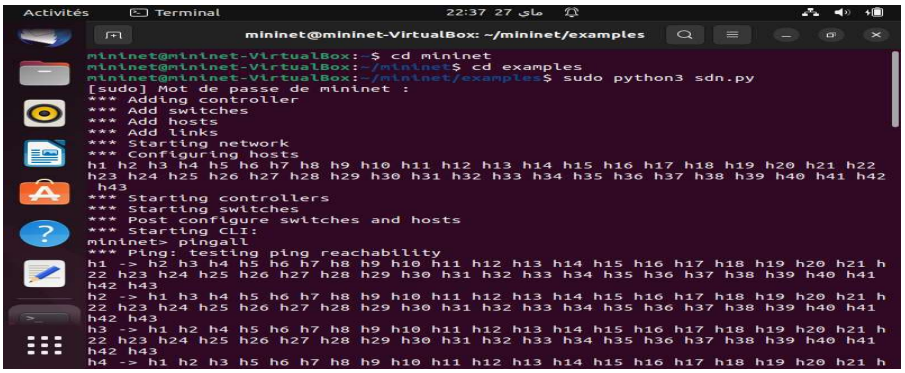


Figure 5.22 : Ping 1 avant l'ajout des règles firewall

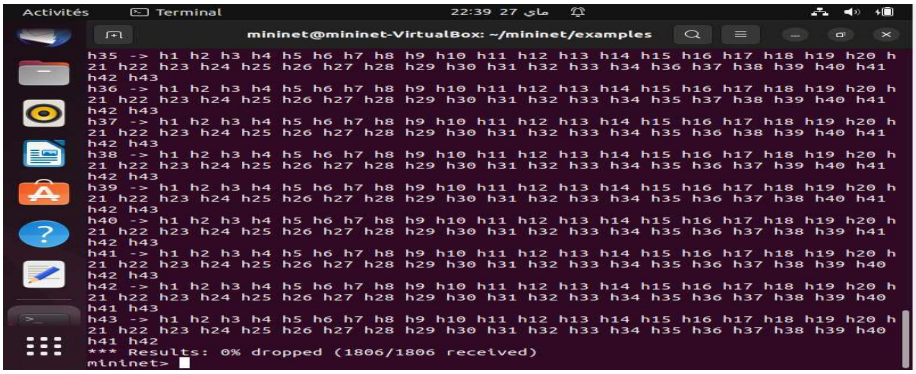


Figure 5.23 : Ping 2 avant l'ajout des règles firewall

3. Ensuite nous avons modifié les règles du Firewall sur firewall.py comme suit

- h1 et h2 se bloquent
- h2 et h4 se bloquent
- h2 et h7 se bloquent
- h3 et h8 se bloquent
- h18 e h6 se bloquent
- h4 et h20 se bloquent
- h10 et h12 se bloquent
- h30 et h9 se bloquent
- h40 et h5 se bloquent
- h2 et h24 se bloquent
- h3 et h13 se bloquent
- h4 et h16 se bloquent
- h5 et h32 se bloquent
- h6 et h11 se bloquent
- h7 et h12 se bloquent

La figure (5.24) ci-dessous représente l'ajout des règles Firewall.

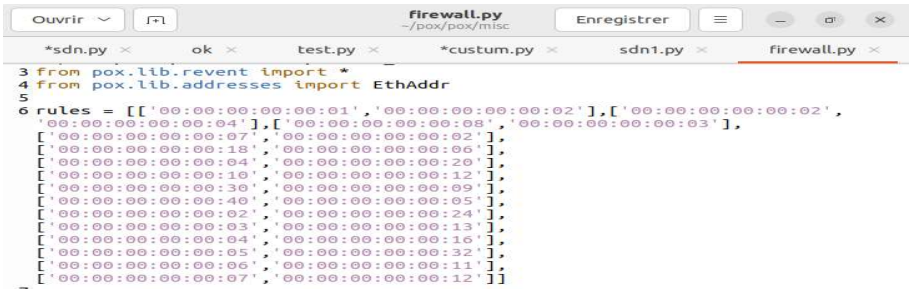


Figure 5.24 : Règles firewall ajouter

4- nous modifions le contrôleur par défaut par Remonte contrôleur sur les trois contrôleurs POX comme illustré ci-dessous 5.25 :

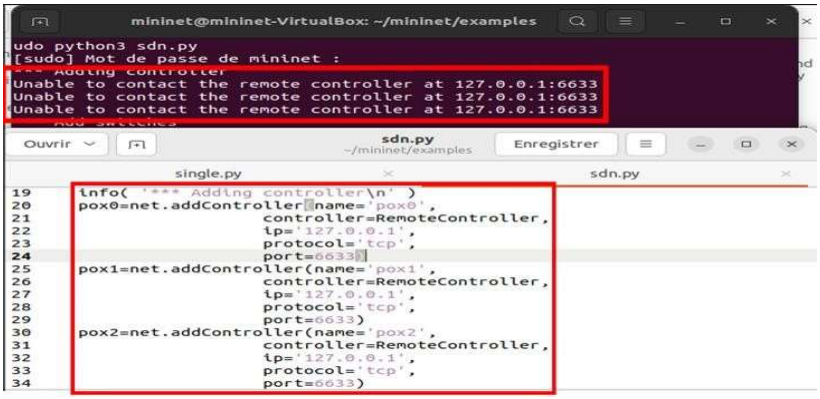


Figure 5.25 : Execution après modification du contrôleur

Méthode 1 :

Nous avons exécuté la commande pingall pour vérifier le fonctionnement de notre firewall sur notre topologie personnalisée qui est illustrée dans la **figure (5.26)**.

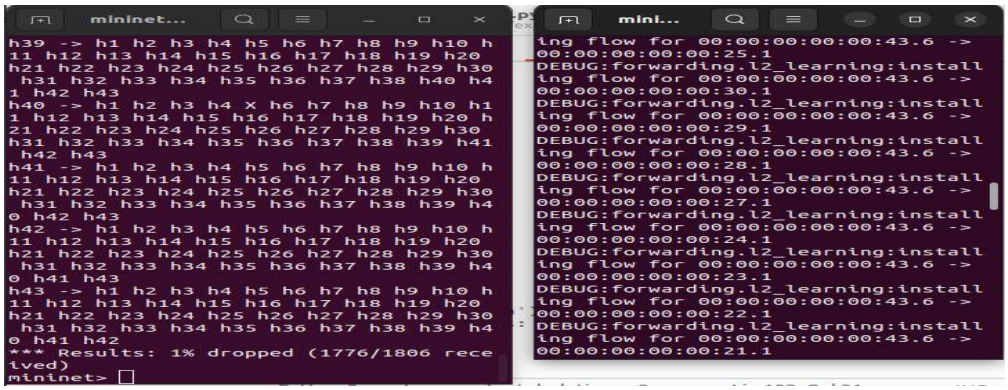


Figure 5.26 : Fonctionnement réel de firewall

Méthode 2 :

Nous faisons un ping entre chaque deux hosts, comme montre les **figures (5.27) (5.28), (5.29), (5.30) et (5.31)**.

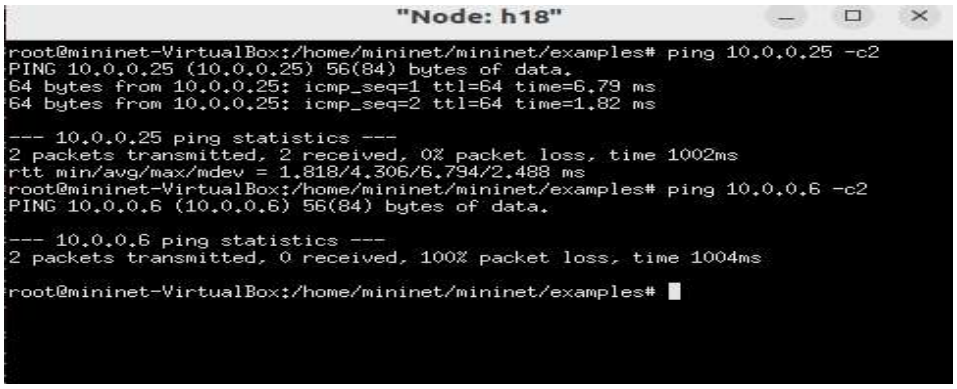


Figure 5.27 : Ping entre h18 et h25 et entre h18 et h6

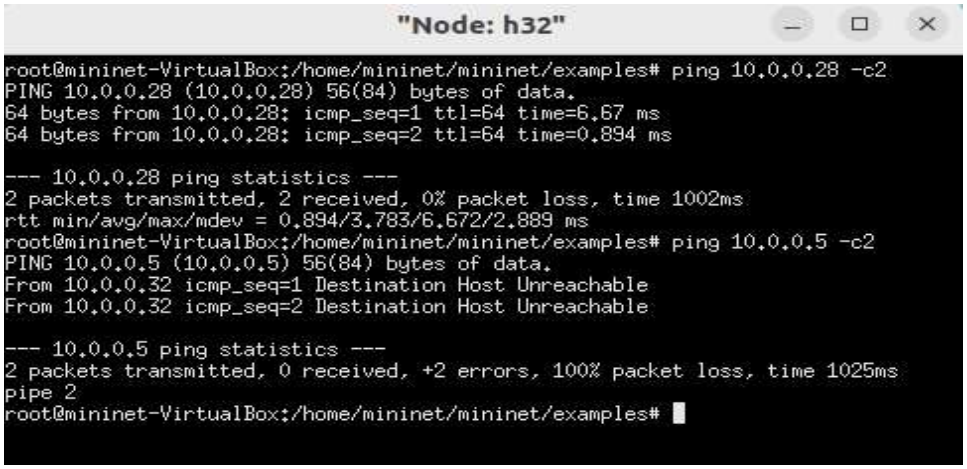


Figure 5.28 : Ping entre h32 et h28 et entre h32 et h5

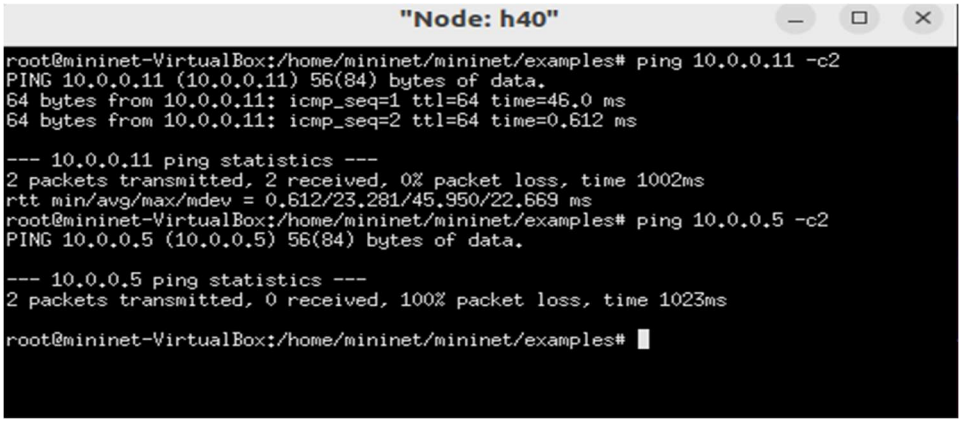


Figure 5.29: Ping entre h40 et h11 et entre h40 et h5

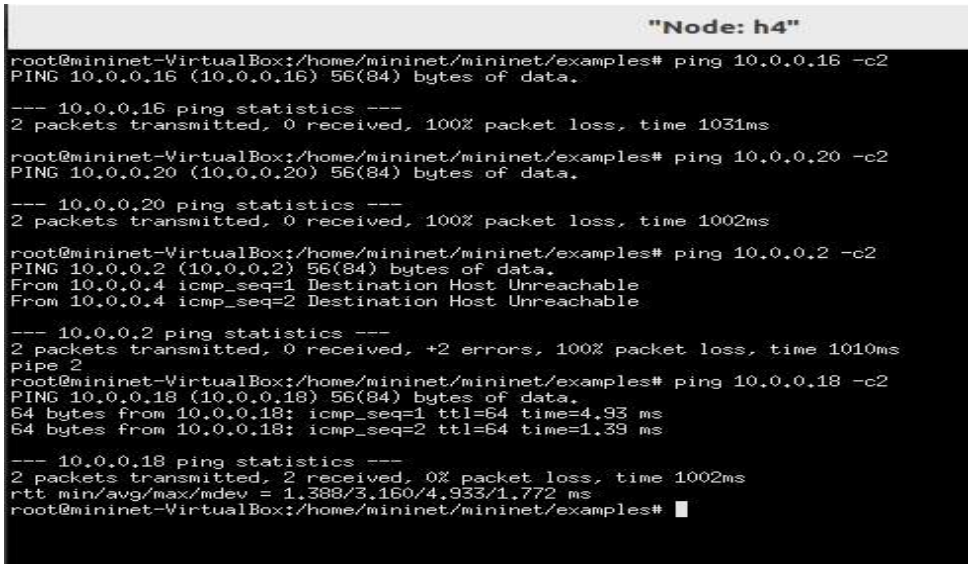


Figure 5.30 : Ping entre h4 et h16 et entre h4 et h20 et entre h4 et h2 et entre h4 et h18

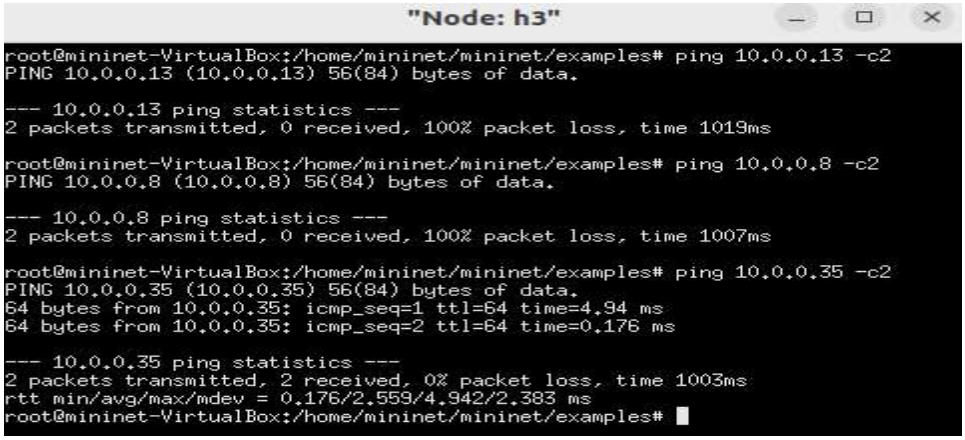


Figure 5.31 : Ping entre h3 et h13 et entre h3 et h8 et entre h3 et h35

Discussion

Selon les résultats des règles d'implémentation de notre Firewall de scénario 2, nous avons affirmé que h1 et h2, h2 et h4, h3 et h8, h2 et h7, h18 e h6, h4 et h20, h10 et h12 h30 et h9, h40 et h5, h2 et h24, h3 et h13, h4 et h16, h5 et h32, h6 et h11, h7 et h12 n'ont jamais pu se connecter, aussi nous voyons qu'il y a une perte de paquets de 1% acceptable.Donc nous avons réussi la mise en place de notre Firewall-SDN pour le scénario 2.

5.4 Conclusion

Dans cette deuxième partie de notre travail pratique nous avons construit une solution SDN de sécurité Firewall de couche 2 À l'aide du contrôleur POX. Nous avons aussi pu gérer les interrupteurs en utilisant POX ainsi que des règles décrivant les attributs de la couche 2 pour autoriser ou interdire la communication entre hôtes.

Actuellement, l'installation des Firewall-SDN physiques peut être coûteuse en termes de matériel. Donc dans cette partie nous avons implémenté un Firewall uniquement sur un logiciel. Cela peut être un futur projet qui aide les entreprises pour la gestion des réseaux SDN.

Conclusion Générale

Conclusion générale

Le SDN (*Software-Defined Networking*), c'est une technologie cruciale pour le moment, L'automatisation de la configuration des équipements est un élément fort et essentiel du SDN, d'où la programmation des réseaux nécessite de bâtir des nouvelles architectures utilisant de nouvelles technologies. Au lieu d'être configuré et opéré par un humain, le réseau devra être géré par des applications. Les applications peuvent dialoguer au travers d'interfaces avec les équipements réseaux ou au travers de contrôleurs réseaux (contrôleurs SDN) offrants des fonctionnalités généralement plus avancées et permettant d'offrir un niveau d'abstraction intéressant.

Dans ce mémoire l'évolution de la performance de certains contrôleurs OpenFlow exécutant une application de communication d'apprentissage a été réalisée ainsi que des analyses pour la technologie SDN (réseau définie par logiciel) et le protocole OpenFlow. Au début du mémoire, nous avons fait une étude de base sur SDN pour explorer cette nouvelle technologie, en utilisant différentes manières associées à des différentes technologies.

Ensuite nous avons étudié l'architecture du protocole OpenFlow pour que nous pouvons comparer certain contrôleurs OpenFlow sont POX et RYU, puis nous avons démontré l'efficacité du contrôleur choisi POX en testant ses performances dans le domaine de la sécurité et de la transmission du trafic.

Enfin en réalisant une solution de système de sécurité Firewall dans un environnement SDN, ce qui nous a permis de prouver la fiabilité ainsi que l'efficacité de réseaux SDN.

Les résultats de notre projet étaient satisfaisants, et nous pouvons dire que nous avons atteint nos objectifs qui sont :

- Surpasser les défauts des réseaux traditionnels.
- Réaliser une solution d'un système de sécurité Firewall basé sur un réseau SDN, sans avoirà utiliser des équipements chers et des protocoles complexes.

En résumé, nous avons effectué notre stage de fin d'études de master réseaux et télécommunications, pendant 2 mois d'où on a pu mettre en pratique nos connaissances

théoriques acquises durant notre formation à l'université de M'HAMED BOUGARA BOUMERDES, tout en étant confronté aux difficultés réelles du monde du travail

Nous avons donc pu s'adapter à chaque situation afin d'accomplir au mieux la tâche qui nous a été donné, ce stage nous a donc été profitable sur le plan relationnel.

Cependant, Au cours de cette période nous avons pu appréhender les difficultés à concevoir,administrer et maintenir un réseau où nous avons été confrontés à des mises en situation qui demandaient une très grande réactivité et une réflexion.

D'un point de vue personnel, nous avons été séduites par une technologie nouvelle et d'avenir. Bien que les mécanismes furnt difficiles à assimiler et les outils très nombreux, la satisfaction de les voir fonctionner ensemble a pris le pas sur les difficultés rencontrées. A postériori nous sommes satisfaites d'avoir choisi ce projet même si nous savons que rien n'est acquis avec un tel sujet en constante mutation.

Pour conclure, le développement d'un tel projet n'est jamais totalement achevé et certainesIdées n'ont pas pu être réalisées à cause de contraintes de temps. C'est dans ce sens et afin d'améliorer ce présent travail que nous proposons de :

- Déployer le SD-WAN
- Implémenter une application de gestions des VLANs dans l'environnement SDN.

Bibliographies

(Aichaoui, 2018) A. Aichaoui, Y. Aitbelkacem , **Etude et implémentation d'une architecture SDN LAN,** Mémoire de Master, université Mouloud MAMMERI de Tizi-Ouzou, ***Soutenu le : 04/07/2018***

(Beroual, 2020) Z.beroual A.Dendane, **Etude et mise en œuvre d'une solution SDN basée sur le contrôleur Floodlight**, Mémoire licence, Institut National de la Poste et des Technologies de L'information et de la Communication, ***2020***.

(Bouida, 2017) H. Bouida, **Etude et mise en oeuvre d'une solution SDN : Application de Gestion de VLANs,** Mémoire de Master, Université Abou Bakr Belkaid– Tlemcen, ***2017.***

(BOURSAS, 2017) H. BOURSAS, **planification et optimisation radio GSM**, rapport de stage, universty des freres mentouri constantin, ***2017***

(Choukri, 2019) I. Choukri, Mohammed Ouzzif, Khalid Bouragba. **Software Defined Networking (SDN): Etat de L'art. Colloque sur les Objets et systèmes Connectés**, Ecole Supérieure de Technologie de Casablanca (Maroc), Institut Universitaire de Technologie d'Aix-Marseille (France), ***Jun 2019.***

(IDOUDI, 2014) K. IDOUDI, **Implémentation d'un plan de contrôle unifie pour les réseaux multicouches IP/DWDM**, Rapport de projet, Université du Québec à Montréal, ***2014.***

(LAMARA, 2018) S. OULD LAMARA, M. TAKILT, **Implémentation du SDN dans une structure IP/MPLS**, Mémoire de Master, Université de Mouloud Mammeri de Tizi-Ouzou, ***2018,***

(Meridji, 2019) R.Meridji, I.Meridji, **Etude Exploratoire Et Mise En Œuvre Des Solutions Basées Sur Sdn Pour Groupe Sonelgaz**, Mémoire Master, université Saad Dahleb-Blida, ***2019.***

(Mazzi, 2019) M. Mazzi, Y.Derbouche**, (route-translator) une solution pour le déploiement du routage IP dans les reseaux SDN**, Mémoire de Master, centre Universitaire abdelhafid boussouf Mila***,2019***

(OULD LAMARA, 2018) S. OULD LAMARA, M. TAKILT***, ' implémentation du SDN dans une structure IP/MPLS',*** Mémoire de Master, université de mouloud Mammeri de Tizi-Ouzou, ***présenter en 2018,***

(OUMEDDI, 2016) A. OUMEDDI I. TOURI, **Installation et mise en œuvre du SDN-IP sur un système ONOS d'un réseau SDN pour assurer le routage inter AS,** Mémoire Master, Institut National de la Poste et des Technologies de l'information et de la communication, ***2016.***

(PARADIS, 2014) T. PARADIS, **Software-Defined Networking**, Mémoire Master, Université de Stockholm, Sweden, ***2014,***

(PARKOO, 2015) K.M. PARKOO, **La qualité de service au niveau IAAS: vers l'utilisation du concept software-defined networking (SDN)**, Mémoire Master , Amadou Hampâté-Bâ de Dakar, ***2015.***

(Semma, 2017) H.Semma, **Conception et implimentation d'une solution d'authentification forte <<PKI>> afin de renforcer la politique de controle d'accès d'Algérie Télécom**, mémoire licence, Institut national de la poste et des technologies de l'information et de la communication, ***2017.***

(ZERTAL, 2020) S. ZERTAL, Cours Sur ***Le Cloud et la Virtualisation,*** Mémoire Master, Université Larbi Ben M'hidi- Oum El Bouaghi, ***2020.***

Webographie

(Alibaba, 2022) Alibaba cloud, ***'Avantages de la virtualisation'***, 2009-2022, https://www.alibabacloud.com/fr/knowledge/what-is-virt, ***Consulté le : 15/03/2022.***

(Astuces, 2022) Astuces & Aide Informatique,***' les avantages de cloud computing'***, publie le 25/03/2019, mis à jour le 02/05/2022, **https://www.astuces-aide-informatique.info/6840/cloud-definition,** ***Consulté le : 16/03/2022.***

(Arizona, 2018), Arizona State University, *'POTENTIAL OF SDN-BASED FIREWALL', 2018*, https://adamdoupe.com/publications/challenges-sdn-firewalls-sdnnfv2018.pdf , ***Consulté -le 9/04/2022.***

(Bastien, 2019) L.Bastien, ***'Virtualisation : qu'est-ce que c'est et à quoi ça sert ?'***,13 mai 2019 https://www.lebigdata.fr/virtualisation-definition, ***Consulté le:17/03/2022.***

(Carlin, 2021) N.Carlin, ***'Virtualisation vs Cloud : la meilleure solution pour votre entreprise'***, 8 décembre 2021 **https://www.parallels.com/blogs/ras/virtualization-vs-cloud/?fbclid=IwAR3vBeSwoPd2rOy2tETY0aCimhCB9IlGlagg-XETpgPDOpp7UH2f4Ux2pXk,** ***Consulté le : 16/03/2022.***

(Canada, 2020) canada.ca, centre canadien pour le cyber sécurité, 'La virtualisation de votre infrastructure (ITSAP.70.011)', publie le **2020,** https://cyber.gc.ca/fr/orientation/la-virtualisation-de-votre-infrastructure-itsap70011,
Consulté le : 19/03/2022.

(Citrix, 2022) Citrix, 'Usages de la virtualisation', 1999-2022, https://www.citrix.com/frc-fr/solutions/vdi-and-daas/what-is-virtualization.html,
Consulté le : 15/03/2022.

(Ccna R/S, 2016) Ccna R/S, *'Examen des plans architecturaux'*, Publié le 26/06/2016, https://www.kwtrain.com/blog/software-defined-networking-sdn-simplified,
Consulté le : 15/03/2022.

(DELL, 2011) *DELL* ***,‘ Éléments de réflexion sur la conversion et le déploiement d'une solution de virtualisation dans les datacenters de PME ‘****, 05/01/2011* https://i.dell.com/sites/content/business/smb/sb360/fr/Documents/wp-workshopplus-fr.pdf et ESET & Blue Bears IT, 21 octobre 2021https://bluebearsit.com/machine-virtuelle-comment-camarche/?fbclid=IwAR3SQBZtBIGoB7tSTiK_AACmw3z9lx2kaeWZ4L3Tp55Rndy8iEG1OBuApK0, ***Consulté le : 19/03/2022.***

(Erell, 2020) Erell Le Gall, ***‘Qu'est-ce que la virtualisation en informatique ?’***, publie le 2020, https://blog.hubspot.fr/marketing/virtualisation-informatique,
Consulté le : 18/03/2022.

(Emilie, 2016) Emilie, ***‘le fonctionnent de cloud computing’***, Publié le 25-05-2016, https://www.openhost-network.com/blog/definition-technologie-cloud/,
Consulte-le : 22/03/2022.

(EFORT, 2016) EFORT**,** ***‘Le Protocole OpenFlow dans l’Architecture SDN Software Defined Network’*****, 2016,** https://dl.ummto.dz/bitstream/handle/ummto/6707/AichaouiAnis_AitbelkacemYanis.pdf,
Consulte-le: 20/03/2022.
(Ijiert, 2021) Ijiert, ***‘Firewall’***, Publié le 09/09/2021, https://repo.ijiert.org/index.php/ijiert/article/download/2899/2544/5304 ,
Consulte-le : 16/04/2022
(Id Excel, 2017) Idexcel Technologies, ***‘le rôle de la cloud computing’***, Posté sur13 novembre 2017, https://www.idexcel.com/blog/top-roles-of-cloud-computing-in-iot/,***Consulté le 20/03/2022.***

(Kaur,2014) S. Kaur, Japinder Singh, N. Ghumman***, ‘Network Programmability Using POX Controller’***, Publié en 2014**. https://www.semanticscholar.org/paper/Network-Programmability-Using-POX-Controller-Kaur-Singh/39baa5b6be6e0966acc519d7d577e1f6ef0c8bea, *consulte-le : 14/05/2022***

(Lucas, 2010) M. Lucas Nussbaum, ***‘les services de cloud’*****,** publies-en 2009/2010computing, https://members.loria.fr/lnussbaum/files/ptasrall2010-cloud-computing-rapport.pdf,
Consulte-le : 23/03/2022.

(Microsoft, 2022) Microsoft, ***'les types de cloud computing'***, publie en 2022, https://azure.microsoft.com/fr-fr/overview/what-is-cloud-computing/#cloud-deployment-types, ***Consulté le : 25/03/2022.***

(Mininet, 2022) Mininet, ***'Mininet Overview'***, publie en 2022, http://mininet.org/overview/?fbclid=IwAR1VBJDcRFoImWjDhNd2x4-gRs-Q82mQf8jZXnIt2mvqS_g4cGk2nmdljs#:~:text=Mininet%3A,are%20repeatable%20and%20easily%20packaged, ***Consulte-le : 03/05/2022.***

(Odjel, 2021) S.odjel, ***'le passage de la virtualisation vers le cloud computing'***, publie le 11 octobre 2021, **https://www.heptabit.at/blog/cloud-migration/virtual-machine-migration-in-cloud-computing?fbclid=IwAR18FdSZUXxyF-TznNNsycm7aNqD17YfouvCIEXvCLSmM-8mGr011sPvgQM,** ***Consulté le : 24/03/2022.***

(Open v switch, 2016) Open v switch***, 'Production Quality, Multilayer Open Virtual Switch'***, publie en 2016, https://www.openvswitch.org/, ***consulte- le : 03/05/2022***

(Openvswitch,2022) openvswitch,https://www.openvswitch.org/support/dist-docs/ovs-vsctl.8.txt, ***consulter le 14/05/2022.***

(Prosica) prosica, ***'Les réseaux SDN - Software Defined Network'*** **https://prosica.fr/blog/118-les-reseaux-sdn-software-defined-network.html,** ***Consulte-le :20/03/2022***

(Redhat,2018), Redhat***, 'Comprendre la virtualisation',*** 19 mars 2018 ***,https://www.redhat.com/fr/topics/cloud-computing/cloud-vs-virtualization , Consulté le : 16/03/2022.***

(Strauss, 2017) A.Strauss ***,'Associer le SDN au cloud pour une gestion plus efficace du réseau',*** Mis à jours 14 novembre 2017, **https://www.hebergeurcloud.com/associer-sdn-cloud-gestion-plus-efficace-reseau/?fbclid=IwAR3ZADS59FwQhU41chhYdF1CzinVdEjCtAAdCqzITkwQcUbF3dwXp_O4JiA#:~:text=L'utilisation%20d'une%20approche,les%20principaux%20probl%C3%A8mes%20de%20s%C3%A9curit%C3%A9 ,Consulte-le 23/03/2022**

, Consulte-le :23/04/2022

(Studios, 2013) Studios SDxCentral**, *'Qu'est-ce qu'OpenFlow ? Définition et relation avec le SDN'*,** publie le 26 août 2013, **https://www.sdxcentral.com/networking/sdn/definitions/what-is-openflow/**
, Consulte-le 24/03/2022
(Studios SDxCentral, 2016) 'Studios SDxCentral, 'What Is a Ryu Controller ?', Publié le 23 juin 2016, https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/what-is-sdn-controller/openflow-controller/what-is-ryu-controller/, ***consulte-le :10/05/2022.***
(Stéphane, 2017), Stéphane LITKOWSKI, publier le 10 avril 2017, https://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-information-th9/administration-de-reseaux-applications-et-mise-en-oeuvre-42481210/software-defined-network-te7609/conclusion-te7609niv10004.html, ***consulter le 10/05/2022.***

(Vahatrainiravo,2015) R. Vahatrainiravo,***'sécurité d'accès du cloud computing en utilisant SSH avec openneblua'*** Mémoire Master, Université D'Antananarivo, 2015, http://biblio.univ-antananarivo.mg/pdfs/razafinimaroVahatrainiravo_ESPA_MASTER_15%20.pdf, ***Consulté-le : 17/03/2022.***

(Wapiti,2010) wapiti**, 'OpenFlow',** publie le 2010, **http://wapiti.enic.fr/commun/ens/peda/options/ST/RIO/pub/exposes/exposesrio2009-ttnfa2010/cloarec-ferreira/controller.html ,*Consulte-le 25/03/2022.***

Annexe A : Sommaire du Stage

Du 28/02/2022 au 28/04/2022, nous avons effectué un stage en direction générale d'Algérie Télécom (situé à Mohammedia, Alger), lors de cette formation dans le département IP/MPLS nous avons pu étudier la technologie SDN qui est actuellement à l'étude en Algérie Télécom.

Plus largement, dans notre période de stage, l'ingénieur de l'encadrement Madame LILA BENSLAMA, nous a envoyés dans deux agences d'Algérie Télécom chargées de l'étude et de la construction de la plateforme SDN.

Nous avons mené un stage d'étude du SDN en environnement réel pendant une semaine dans la première agence d'Algérie Télécom située à Kouba, qui est chargée d'établir la plateforme SDN avec des équipements fournis par Huawei, pour faire fonctionner le contrôleur NCE.

Notre deuxième stage d'apprentissage de la technologie SDN en milieu réel pendant 10 jours à l'Agence d'Algérie Télécom de Bordj El-Kifan, cette agence utilise des équipements de réalisation d'une plateforme SDN fournis par Juniper, pour utiliser du contrôleur NORTHSTAR.

Annexe B : Matériels plate-forme SDN d'Algérie Télécom

La plateforme SDN d'Algérie Télécom est basée sur :

- NORTHSTAR Juniper qui utilise deux contrôleurs le contrôleur principal est situé à Setif et son secoure qui est situé à Bordj El-Kifan
- NCE Huawei qui utilise aussi deux contrôleurs, le contrôleur principal est situé à Kouba et le second qui est situé en Oran.

1. *Solution SDN Juniper*

Comme la plupart de ses concurrents, Juniper Networks développe toutes sortes de composants logiciels, dont un certain nombre axé sur l'automatisation des réseaux, afin de répondre aux besoins changeants des entreprises et accélérer la prestation de services.

Les principaux équipements JUNIPER utilisés pour la préparation de la réalisation de la plate-forme d'Algérie Télécom actuellement sont :

A- La série des serveurs

- ***Serveur NEC Express5800/R120h-1M***

L'Express5800/R120h-1M, un serveur rack 1U à double socket haute densité, offre des performances, une évolutivité et une fiabilité exceptionnelles.

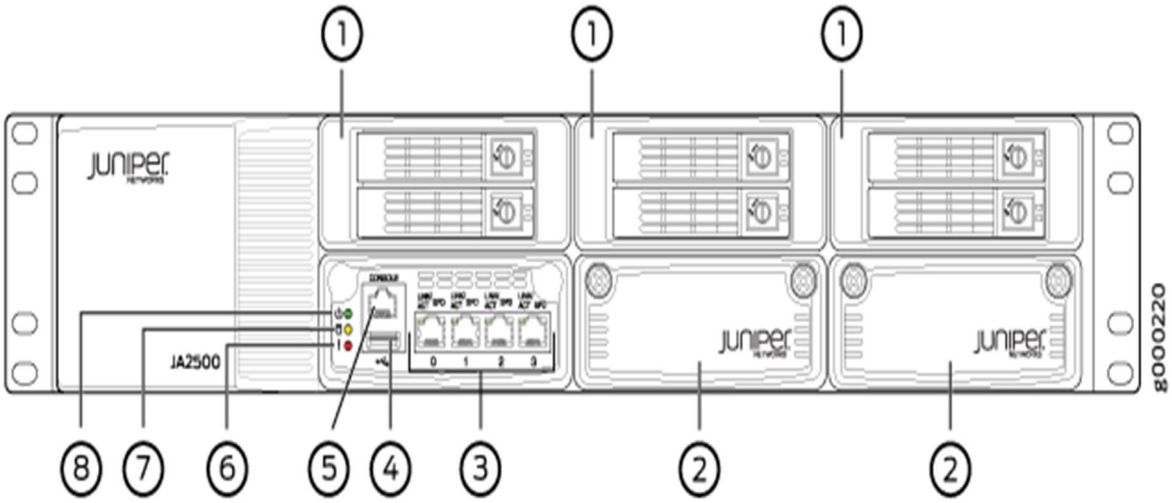
Caractéristique	R120h-1M
Facteur de forme / hauteur	Rack 1U
Nombre de processeurs	1 à 2
Processeur	Processeur Intel® Xeon® Bronze. Processeur Intel® Xeon® Argent. Intel® Processeur Xeon® Gold. Processeur Intel® Xeon® Platinum.
Type de mémoire	DDR4-2666 ECC RDIMM / LRDIMM.
Emplacements de mémoire	24
Mémoire maximale	3 To.
Type de stockage	Disque dur SAS 2,5 pouces enfichable à chaud. Disque dur SATA 2,5 pouces enfichable à chaud Disque SSD SAS 2,5 pouces Enfichage à chaud SSD SATA 2,5 pouces. Disque dur SAS Near Lineb3,5 pouces enfichable à chaud. Disque dur SATA 3,5 pouces enfichable à chaud.
Nombre maximal de baies de lecteur internes	2,5 pouces : 11 3,5 pouces : 4
Stockage interne maximal	42 To
Baies multimédias amovibles	1 baie pour lecteur de disque optique.
Logements d'extension	1 PCIe x16 Gen 3. 1 PCIe x8 Gen 3. 1 PCIe x8 Gen 3 pour une carte RAID dédiée. 1 PCIe x8 Gen 3 pour un contrôleur LOM dédié. * La combinaison de logements change en installant des cartes de montage en option.
Interface réseau	4 1000BASE-T plus 1 1000BASE-T pour la gestion
Alimentation redondante	Standard, branchement à chaud
Ventilateur de refroidissement redondant	Standard, branchement à chaud
Alimentations	Bloc d'alimentation enfichable à chaud certifié 80 PLUS® Platinum 500 W/800 W AC100-120 V/200-240 V±10 %, 50/60 Hz ± 3 Hz Bloc d'alimentation enfichable à chaud certifié 80 PLUS® Titanium 800 W 80 PLUS®/1 600 W 80PLUS® Platinum 60Hz±3Hz.
Interface	1 VGA, 1 série (en option), 3 à 4 USB 3.0 (plus 2 USB 3.0 internes), 4 LAN, 1 LAN de gestion
Dimensions (L x P x H)	2,5 pouces : 434,6 x 707,0 x 42,9 mm / 17,1 x 27,8 x 1,7 pouces 3,5 pouces : 434,6 x 749,8 x 42,9 mm / 17,1 x 29,5 x 1,7 pouces
Conditions de température et d'humidité (sans condensation)	En fonctionnement : 10 à 35 °C / 50 à 95 °F, 8 à 90 % À l'arrêt : -30 à 60 °C / -22 à 140 °F, 5 à 95 %
Systèmes d'exploitation et logiciels de virtualisation	Microsoft® Windows Server® 2012 R2 Standard / Centre de données. Microsoft® Windows Server® 2016 Standard / Centre de données. Red Hat® Enterprise Linux® 6. Red Hat® Enterprise Linux® 7. VMware® ESXi™ 6.0. VMware® ESXi™ 6.5.

- ***Serveur JA2500***

Juniper Networks junos Space Appliance est un dispositif matériel dédie conçu pour fournir la puissance de calcul et répondre aux exigences spécifiques.

Les Appliance Junos Space sont actuellement disponibles en un seul modèle : JA2500.

- ***Panneau avant de serveur JA2500***

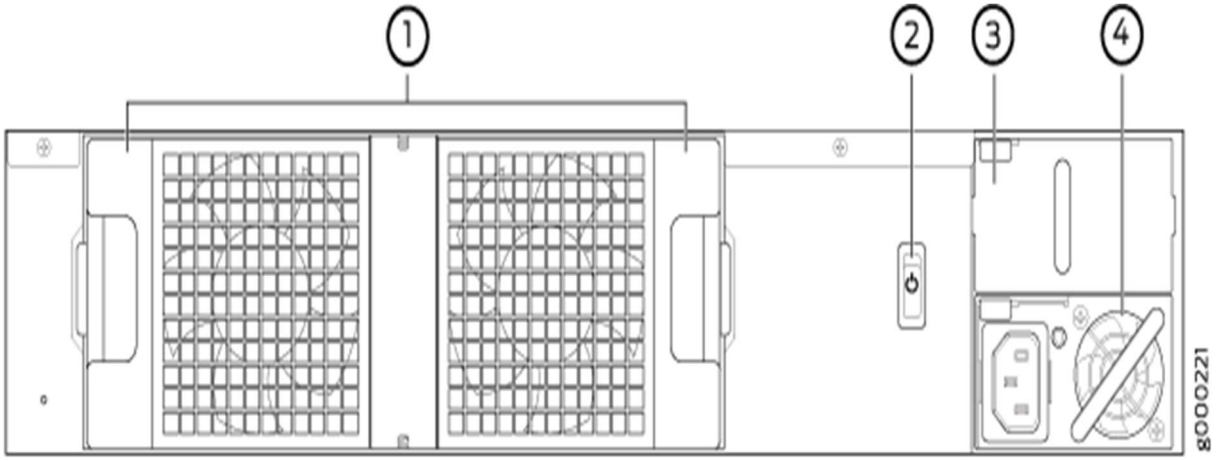


1- Disque dur	5-Port console
2-Fente d'extension IOC	6-Voyant de panne matérielle
3-Ports réseau (Ethernet)	7-Voyant d'activité du disque dur
4-port USB	8-Voyant d'alimentation

Caractéristiques panneaux avant	JA2500
Disques durs	L'Appliance JA2500 dispose de six disques durs de 1 To remplaçables à chaud dans une configuration RAID 10. Les disques durs sont numérotés (de 0 à 5) comme suit : • Slot 3 (en haut à gauche) et Slot 0 (en bas à gauche). • Emplacement 4 (en haut au milieu) et emplacement 1 (en bas au milieu). • Emplacement 5 (en haut à droite) et emplacement 2 (en bas à droite).
Logements d'extension de carte d'E/S (IOC)	L'extension IOC n'est actuellement pas prise en charge.
Connexions réseau	Quatre ports Ethernet RJ-45 10/100/1000, étiquetés de 0 à 3 de gauche à droite.
Port USB	Un port USB
Port console	Un port de console RJ-45
Voyants du châssis	Les voyants du châssis suivants, situés à côté du port de console, sont présents sur l'appliance : Voyant de panne matérielle (rouge), qui indique qu'une alarme de ventilateur, d'alimentation ou de température s'est produite. Voyant d'activité du disque dur (jaune), qui indique que le disque dur est en cours d'utilisation. Voyant d'alimentation (vert) qui indique que l'appliance est sous tension.
Voyants du port Ethernet (LAN)	Les voyants suivants sont présents au-dessus de chaque port Ethernet : • LED de liaison/activité (verte), qui indique la liaison (allumée) ou l'activité de la liaison (clignotant) Voyant de vitesse, qui indique la vitesse de liaison : • Désactivé—10 Mbit/s • Vert—100 Mbit/s • Jaune–1000 Mbps ou 1 Gbps
Voyants du disque dur	Outre les voyants du châssis de l'Appliance, il y a deux voyants sur chaque disque dur : • Voyant d'activité du disque dur (vert), qui indique l'activité du disque. • LED de panne de disque dur (rouge), qui indique une panne de disque (allumée) ou une reconstruction de disque (clignotant).

Caractéristiques panneaux arrière	JA2500
Ventilateurs de refroidissement	L'Appliance JA2500 dispose de deux ventilateurs de refroidissement remplaçables à chaud qui fournissent le flux d'air requis pour refroidir l'Appliance.
Interrupteur	L'interrupteur d'alimentation de l'appareil est utilisé pour allumer ou éteindre l'appareil.
Emplacement du module d'alimentation redondante.	Un emplacement de module d'alimentation vide est fourni afin qu'un deuxième module d'alimentation puisse être ajouté, pour la redondance, si nécessaire.
Module d'alimentation	Un seul module d'alimentation AC est livré avec l'Appliance et alimente l'Appliance. Cependant, l'Appliance peut également fonctionner sur DC en utilisant le module d'alimentation DC pour l'Appliance JA2500
Voyant du module d'alimentation	Une LED est présente sur le module d'alimentation : • Le vert indique que le module d'alimentation alimente l'appareil. • Orange indique que le module d'alimentation est allumé mais qu'il n'alimente pas l'Appliance (mode veille).

- ***Panneau arrière de serveur JA2500***



1-Ventilateurs de refroidissement	3-Emplacement du module d'alimentation redondante
2-Interrupteur	**4**-Module d'alimentation

B- La série des switchs

- ***EX2300***

Le commutateur Ethernet EX2300 de Juniper Networks est une solution d'entrée de gamme, économique et autonome, idéale pour les déploiements de couche d'accès dans les filiales et bureaux distants, et pour les réseaux des campus d'entreprise.

Caractéristique	EX2300
Densité de ports	Jusqu'à 48 x 1 GbE et 4 SFP/SFP+ 10 GbE (liaisons montantes)
Facteur de forme	1 U
Alimentation	PoE/PoE+ jusqu'à 30 W par port
Capacité de commutation	176 Gbits/s

- ***SRX1500***

Une passerelle de services de sécurité qui protège les campus d'entreprise critiques, les sièges sociaux régionaux et les réseaux de centres de données.

Caractéristiques	SRX1500
Haute performance	Jusqu'à 9 Gbit/s de performances de pare-feu
Expérience utilisateur final de haute qualité	Visibilité et contrôle des applications
Protection contre les menaces	IPS, antivirus, anti-spam, filtrage Web amélioré, Juniper Advanced Threat Prevention Cloud, Encrypted Traffic Insights, Threat Intelligence Feeds et Juniper ATP Appliance
Services de mise en réseau de qualité professionnelle	Routage, commutation et câblage sécurisé
Hautement sécurisé	VPN IPsec, accès à distance/VPN SSL, démarrage sécurisé
Grande fiabilité	Cluster de châssis, alimentation redondante
Facile à gérer et à faire évoluer	Interface graphique intégrée, directeur de la sécurité
Coût total de possession inférieur	Système d'exploitation Junos

C- Métro switch EX8208

Le commutateur EX8208 est un système modulaire qui offre une haute disponibilité et une redondance pour tous les principaux composants matériels, y compris les moteurs de routage, la matrice de commutation, le plateau de ventilation (ventilateurs redondants) et les blocs d'alimentation.

Caractéristiques	EX8208
Châssis	• 4 RU ; 21 po (53 cm) de profondeur ; 17,25 po (43,8 cm) de large. • 8 emplacements d'E/S dédiés. • Capacité de fond de panier de 6,2 Tbit/s. • Données, contrôle et gestion dédiée avions. • Panneau LCD pour la surveillance du système.
Énergie	• Efficacité énergétique : jusqu'à 195 000 paquets par seconde par watt. • 6 alimentations à répartition de charge. • Capacité de puissance maximale de 10 000 W. • Options 220 V CA, 110 V CA et -48 V CC pour N+1 ou redondance N+N.
Refroidissement	• Ventilateurs et contrôleurs redondants à vitesse variable. • Flux d'air latéral.
Fabrique	• Capacité de structure de 320 Gbit/s (duplex intégral) par logement. • Redondance 2+1 avec double SRE et carte SF. • Transfert à débit de ligne complet avec deux fabriqués en système.
Moteur de routage	• Redondance 1+1. • SRE maître et sauvegarde. • 2 Go de DRAM ; 2 giga-octets de mémoire Flash. • Console + série auxiliaire et Ethernet ports de gestion. • Interface de stockage USB.
Fonctionnement du système	• Junos OS.
Haute disponibilité	Matériel conçu pour un fonctionnement continu : • L'architecture sécurisée et modulaire isole les pannes. • Les plans de contrôle et de transfert séparés améliorent évolutivité et résilience. • Basculement transparent et récupération du réseau. • Commutation gracieuse du moteur de routage (GRES). • Routage sans escale (NSR). • Pontage sans escale (NSB). • Mise à niveau logicielle non-stop (NSSU).
Caractéristiques couche 2	• Trames Jumbo (9216 octets). • 4 096 VLANs. • Protocole d'enregistrement VLAN (GVRP). • VLAN privé (PVLAN). • 802.3ad – Protocole de contrôle d'agrégation de liens (LACP). • 802.1D – Protocol Spanning Tree (STP). • 802.1w - protocol Rapid Spanning Tree (RSTP). • 802.1s – Arbre couvrant plusieurs instances Protocole (MSTP). • protocol Spanning Tree VLAN (VSTP). • Groupe de liaisons redondantes (RTG).
Caractéristiques couche 3	• Routage statique. • RIP v1/v2. • OSPF v1/v2. • Transfert basé sur des filtres. • Protocole de redondance de routeur virtuel (VRRP). • BGP (licence de fonctionnalité avancée). • IS-IS (licence de fonctionnalités avancées). • IPv6 (licence de fonctionnalité avancée). • Détection de transfert bidirectionnel (BFD). • Routeurs virtuels.
Matériel creuser un tunnel	• Tunnels GRE*. • Capacités MPLS (licence de fonctionnalité avancée).
Multidiffusion	• Protocole de gestion de groupe Internet (IGMP) v1/v2/v3. • Surveillance IGMP v1/v2/v3. • Multidiffusion indépendante du protocole PIM-SM, PIMSSM, PIM-DM, MSDP.
Filtres de pare-feu	• Listes de contrôle d'accès d'entrée et de sortie L2-L4 (ACL) : - Port ACL. - ACL VLAN. - ACL de routeur. • Protection contre le déni de service (DoS) du plan de contrôle.
Qualité de services (QoS)	2 000 policiers par châssis : • 8 files d'attente de sortie par port. • Abandon précoce aléatoire pondéré (WRED) Planification. • Tournoi à la ronde pondéré en fonction du déficit (SDWRR) faire la queue. • File d'attente prioritaire stricte. • Classique multidisciplinaire.
La gestion	• Interface de ligne de commande (CLI) Junos OS. • Protocole de gestion Junos XML. • Gestion Web intégrée – Junos la toile. • Prise en charge du gestionnaire de réseau et de sécurité (NSM). • Écran LCD. • SNMP v1/v2/v3. • RAYON. • TACACS+. • Prise en charge MIB étendue. • Analyseur local et distant (mise en miroir). • Protocole de découverte de couche liaison (LLDP). • Solutions d'analyse avancées (AIS).

D- Le routeur MX960

La plate-forme de routage universelle MX960 compatible SDN offre des performances et une évolutivité éprouvée sur le terrain aux plus grands fournisseurs de services, réseaux câblés, mobiles et de centres de données au monde.

Caractéristiques	MX960
Capacité du système	Jusqu'à 12 Tbit/s.
Capacité de matrice de commutation par emplacement	Jusqu'à 1,5 Tbit/s.
DPC et/ou MPC par châssis	Jusqu'à 11.
LMIC et RE par châssis	N / A.
Châssis par rack	3.
Dimensions (L x H x P)	17,37 x 27,75 x 28 pouces (la profondeur totale inclut la mesure standard du gestionnaire de câbles) (44,11 x 70,49 (16 U) x 71,1 cm)
Poids maximal	334 livres / 151,6 kg.
Montage	Avant ou centre
Options d'alimentation	CA/CC : -40 à -72 V CC 100-240 V CA.
Température de fonctionnement	0–104°F (0–40°C) à 6000 pieds.
Humidité	5 à 90 % sans condensation.
Altitude	Aucune dégradation des performances jusqu'à 13 000 pieds/4 000 m.

2- Solution SDN huawei

Huawei innove depuis des années en matière de SDN, à la fois dans le matériel et les logiciels. Ils ont lancé un commutateur programmable, qui a un plan de contrôle programmable en silicium. Ils viennent s'ajouter au logiciel de type SDN pour le rendre programmable.

Les principaux équipements HUAWEI utilisés pour la préparation de la réalisation de la plate-forme d'Algérie Télécom actuellement sont :

A- Métro switch game S12700E-4 cloud Engine

CloudEngine S12700E permet une convergence filaire et sans fil, une ouverture complète et des mises à niveau fluides au niveau de la couche centrale des réseaux de campus haut de gamme.

Caractéristiques	Cloud Engine S12700E-4
Performances de transfer	14 400 Mpps
Capacité de commutation 2	19,2 Tbit/s/28,8 Tbit/s
Emplacements MPU	2
Machines à sous SFU	2
Emplacements pour cartes de service	4
Emplacements de ventilateur	2
Services sans fil	Gère jusqu'à 10 000 points d'accès Contrôle d'accès des points d'accès, gestion des domaines des points d'accès et gestion des modèles de configuration des points d'accès gestion des canaux radio, configuration statique unifiée et gestion centralisée dynamique. Services de base WLAN, qualité de service, sécurité et gestion des utilisateurs. CAPWAP, emplacement des balises/terminaux et spectre une analyse.
iPCA	Collecte de statistiques en temps réel sur le nombre de paquets perdus et le taux de perte de paquets au niveau du réseau et des appareils.
Super tissu virtuel (SVF)	Jusqu'à 10 000 clients (commutateurs d'accès et points d'accès) virtualisés dans un seul appareil. Prend en charge une architecture client à deux couches. Prend en charge les appareils tiers entre le parent SVF et les clients.
VXLAN	Passerelles VXLAN L2 et L3 Passerelles centralisées et distribuées. BGP-EVPN. Configuré via le protocole NETCONF.
Interopérabilité	VBST (compatible avec PVST, PVST+ et RPVST). LNP (similaire à DTP). VCMP (similaire à VTP). Pour des certifications d'interopérabilité détaillées et des rapports de test.

B-Switch S5331

Les commutateurs de la série CloudEngine S5331-H sont de tous nouveaux commutateurs d'accès Gigabit améliorés qui fournissent des ports de liaison descendante électrique entièrement GE et des ports de liaison montante 10GE fixes.

Caractéristiques	CloudEngine S5331-H24T4XC	CloudEngine S5331-H24P4XC	CloudEngine S5331-H48T4XC	CloudEngine S5331-H48P4XC
Port fixe	24 10/100/1000Base-T ports, 4 SFP+ 10GE ports.	24 10/100/1000Base-T (PoE+) ports, 4 10GE SFP+ ports	48 10/100/1000Base-T ports, 4 10GE SFP+ Ports.	48 10/100/1000Base-T(PoE+) ports, 4 10GE SFP+ ports.
Dimensions (W x D xH)	442 mm x 420 mm x 43.6 mm	442 mm x 420 mm x 43.6 mm	442 mm x 420 mm x 43.6 mm	442 mm x 420 mm x 43.6 mm
Hauteur du châssis	1U	1U	1U	1U
Poids du châssis (plein poids de configuration)	8.4kg	8.6kg	8.55kg	8.8kg
Emplacement étendu	Un emplacement étendu, Réservé pour une utilisation future	Un emplacement étendu, Réservé pour une utilisation future	Un emplacement étendu, Réservé pour une utilisation future	Un emplacement étendu, Réservé pour une utilisation future
Tension d'entrée	• Tension nominale plage : 100 V AC à 240 V AC, 50/60 hertz • Tension maximale Portée: − Entrée AC : 90 V AC à 290 V AC, 45 Hz à 65Hz − Haute tension Entrée DC : 190 V DC à 290 V DC	• Tension nominale plage : 100 V AC à 240 V AC, 50/60 hertz • Tension maximale Portée: − Entrée AC : 90 V AC à 290 V AC, 45 Hz à 65Hz − Haute tension Entrée DC : 190 V DC à 290 V DC	• Tension nominale plage : 100 V AC à 240 V AC, 50/60 hertz • Tension maximale Portée: − Entrée AC : 90 V AC à 290 V AC, 45 Hz à 65Hz − Haute tension Entrée DC : 190 V DC à 290 V DC	• Tension nominale plage : 100 V AC à 240 V AC, 50/60 hertz • Tension maximale Portée: − Entrée AC : 90 V AC à 290 V AC, 45 Hz à 65Hz − Haute tension Entrée DC : 190 V DC à 290 V DC
Puissance maximum consommation	114 W	•121 W (sans PDs); • 977 W (avec PDs, PDs : 740 W)	124 W	•132 W (sans PDs); •1750 W (avec PDs,PDs: 1440 W)

Caractéristiques	CloudEngine S5331-H24T4XC	CloudEngine S5331-H24P4XC	CloudEngine S5331-H48T4XC	CloudEngine S5331-H48P4XC
Bruit	•Dans des conditions normales Température (puissance sonore): 57,5 dB(A) • Sous haute Température (puissance sonore): 70,9 dB(A) •Dans des conditions normales Température (pression sonore): 47,5 dB(A)	•Dans des conditions normales Température (puissance sonore): 62.3 dB(A) • Sous haute Température (puissance sonore): 71.8 dB(A) •Dans des conditions normales Température (pression sonore): 52.8dB(A)	•Dans des conditions normales Température (puissance sonore): 57,5 dB(A) • Sous haute Température (puissance sonore): 70,9 dB(A) •Dans des conditions normales Température (pression sonore): 47,5 dB(A)	•Dans des conditions normales Température (puissance sonore): 62.3dB(A) • Sous haute Température (puissance sonore): 71.8 dB(A) •Dans des conditions normales Température (pression sonore): 52.8dB(A)
Température de fonctionnement	• 0-1800 m d'altitude : -5°C à 45°C • 1800-5000 mètres Altitude : Le en fonctionnement Température réduit de 1ºC chaque fois que le augmentation d'altitude	• 0-1800 m d'altitude : -5°C à 45°C • 1800-5000 mètres Altitude : Le en fonctionnement Température réduit de 1ºC chaque fois que le augmentation d'altitude	• 0-1800 m d'altitude : -5°C à 45°C • 1800-5000 mètres Altitude : Le en fonctionnement Température réduit de 1ºC chaque fois que le augmentation d'altitude	• 0-1800 m d'altitude : -5°C à 45°C • 1800-5000 mètres Altitude : Le en fonctionnement Température réduit de 1ºC chaque fois que le augmentation d'altitude
Température de stockage	-40℃～70℃	-40℃～70℃	-40℃～70℃	-40℃～70℃
Humidité relative	5 % à 95 % (sans condensation)	5 % à 95 % (sans condensation)	5 % à 95 % (sans condensation)	5 % à 95 % (sans condensation)
Protection contre les surtensions spécification (service Port)	Mode commun : ±6 kV	Mode commun : ±6 kV	Mode commun : ±6 kV	Mode commun : ±6 kV
Protection contre les surtensions spécification (puissance Port)	• Mode différentiel : ±6kV •Mode commun : ±6 kV	• Mode différentiel : ±6kV •Mode commun : ±6 kV	• Mode différentiel : ±6kV •Mode commun : ±6 kV	• Mode différentiel : ±6kV •Mode commun : ±6 kV
Dissipation de la chaleur	Chaleur de refroidissement par air dissipation, intelligent réglage de la vitesse, et ventilateurs enfichables	Chaleur de refroidissement par air dissipation, intelligent réglage de la vitesse, et ventilateurs enfichables	Chaleur de refroidissement par air dissipation, intelligent réglage de la vitesse, et Ventilateurs enfichables	Chaleur de refroidissement par air dissipation, intelligent réglage de la vitesse, et ventilateurs enfichables

D- Commutateur Huawei CloudEngine 6881-48S6CQ

Il prend en charge des fonctionnalités de centre de données riches et un empilement haute performance, et la direction du conduit d'air peut être sélectionnée de manière flexible

Caractéristiques	CloudEngine 6881-48S6CQ
Ports en aval	48 ports SFP+ 10GE
Ports en amont	6 ports 40/100 GE QSFP28
Capacité de commutation	2,16 Tbit/s
Vitesse de transfert	940 Mpps
Amortir	42 Mo
Fiabilité	Prise en charge du protocole Prise en charge du protocole LACP Micro-segmentation. Détection de transfert de données bidirectionnelle matérielle.
Exploitation et maintenance	Technologie de télémétrie flux net sFlow ESPAGNE+
Caractéristiques du centre de données	Routage et pont VXLAN BGP-EVPN Groupe d'agrégation de liens M-LAG PFC et ECN
Consommation électrique maximale	380W
Source de pouvoir	Courant alternatif : 600 W DC : 1200W -48V Haute tension DC : 1 200 W 380 V
Tension de travail	AC : 90 V à 290 V DC : -38,4 V à -72 V Haute tension CC : 190 V ~ 400 V

E- Routeur HUAWEI NetEngine40E-X8A

La série NetEngine 40E fournit les cartes de ligne de routage 2T de la plus grande capacité du secteur. Combinant des performances avec une faible consommation d'énergie, une technologie innovante de protocole Internet (IP) et des capacités d'évolution rapide, les routeurs NetEngine 40E répondent aux exigences de faible latence et de haute fiabilité des services critiques ainsi que des logiciels de réseau étendu (WAN) matures.

Caractéristiques	NetEngine40E-X8A
Capacité de commutation	7.08 Tbit/s
Performances de transfert	2880 Mpps
Emplacements MPU	2
Machines à sous SFU	3
Emplacements LPU	8
Dimensions (H x L x P)	620 X 442 X 650 millimètre
Poids en configuration complète	- 119 kg (120g) - 136 kg (240g)

E-Serveur TaiShan 200 modèle 2280

Il dispose d'un calcul haute performance, d'un stockage de grande capacité, d'une faible consommation d'énergie consommation, gestion facile et déploiement facile et est idéal pour Internet, stockage distribué, cloud computing, sdn, Big Data et services d'entreprise.

Caractéristiques	TaiShan 200 modèle 2280
Facteur de forme	2U serveur rack
Processeurs	2 Kunpeng 920 processeurs
Mémoire	32 DDR4-2933 emplacements DIMM
Stockage interne	12-modèle d'entraînement : • Avant : 12x3.5 pouces SAS/SATA ou Ssd; Arrière: 4x3.5 pouces SAS/SATA ou Ssd. 25-modèle d'entraînement : • Avant : 25x2.5 pouces SAS/disques SATA ou Ssd ou 16x2.5 pouces Ssd NVMe; arrière: 2x2.5 pouces SAS/SATA ou Ssd.
RAID	RAID 0, 1, 5, 6, 10, 50 ou 60. Super condensateur pour protection contre les pannes de courant.
LOM Ports Réseau	2 flexibles LOM cartes, prenant en charge jusqu'à 8 x GE ports électriques, 8x25 GE/10 ports optiques GE, ou 4 x GE ports électriques + 4x25 GE/10 ports optiques GE
Extension PCIe	Jusqu'à 8 PCIe 4.0x8 ou 3 PCIe 4.0x16 + 2 PCIe 4.0x8 emplacements
Unités D'alimentation	2x1, 500W ou 2,000W remplaçable à chaud AC Psu et redondance 1 + 1
Alimentation	100V à 240V AC; 240V DC
Modules de ventilateur	4 modules de ventilateurs remplaçables à chaud en redondance N + 1
Température de fonctionnement	5 °C à 40 °C (41 °F à 104 °F)
Dimensions (H x L x P)	86.1mm x 447mm x 790mm (3.39. X 17.60. X 31.10.)

F-Passerelle hies engine Eudemon1000E-G15

L'Eudemon1000E-G est un nouveau pare-feu de nouvelle génération de produit, développé par Huawei pour répondre aux besoins des transporteurs, entreprises et les centres de données de nouvelle génération Il fournit des interfaces de plate-forme d'application (API) standard, ainsi que le cloud OpenStack plate-forme, contrôleur SDN pour obtenir des solutions intelligentes pour la sécurité du cloud.

Caractéristiques	Eudemon1000E -G15
Débit du pare-feu[1] (1518/512/64 octets, UDP)	10/10/10 Gbit/s
Latence du pare-feu (64 octets, UDP)	15 µs
Sessions simultanées (HTTP1.1)[1]	6, 000,000
Nouvelles sessions/seconde (HTTP1.1)[1]	200,000
Débit VPN IPsec1 (AES-256 + SHA256, 1420 octets)	10 Gbit/s
Débit d'inspection SSL[2]	3 Gbit/s
Concurrent SSL VPN Users (Default/Maximum)	100/2000
Politiques de sécurité (maximum)	40,000
Pare-feu virtuels	200
Filtrage d'URL Catégories	Plus de 130
Filtrage d'URL : URLs	Une base de données de plus de 120 millions d'URL dans le cloud.
Commentaires automatisés sur les menaces et Mises à jour des signatures IPS	Oui, un centre de sécurité de pointe de Huawei (Http://sec.huawei.com/sec/web/index.do)
Tiers et Open Source ecosystème	Open API pour l'intégration avec des produits third-party, fournissant RESTful et Interfaces NetConf Autres logiciels de gestion third-party basés sur SNMP, SSH et Syslog Co-opération avec des outils third-party, tels que Tufin, AlgoSec et FireMmon Collaboration avec la solution anti-APT
Gestion centralisée	La configuration centralisée, la journalisation, la surveillance et la création de rapports sont effectuées par Huawei eSight et eLog
VLANs (maximum)	4094
Interfaces VLANIF (maximales)	1024

Résumé

La croissance exponentielle des utilisateurs du réseau et leurs demandes de communicationont conduit à un accroissement considérable de la consommation d'énergie dans les infrastructures réseau. Un nouveau paradigme de réseautage appelé Software Defined Networking (SDN) a récemment vu le jour, dans lequel le plan de données, responsable du transfert de paquets est découplé du plan de contrôle responsable de la prise de décision.

Le SDN permet à l'administrateur d'utiliser des équipements moins onéreux et de renforcer son contrôle des flux de trafic du réseau avec l'utilisation du protocole OpenFlow qui a été initialement permet de programmer le plan de contrôle d'un équipement réseau.

Dans ce mémoire, nous avons fait une évaluation des performances de deux contrôleurs SDN, ces contrôleurs sont POX et RYU, nous avons aussi réalisé une solution de sécurité Firewall avec le contrôleur le plus performant dans un environnement SDN.
Mots-clefs: SDN, Contrôleur, OpenFlow, Pox, Ryu, Firewall.

Abstract

The exponential growth of network users and their communication demands have led to a considerable increase in energy consumption in network infrastructures. A new networking paradigm called Software Defined Networking (SDN) has recently emerged, in which the dataplane, responsible for packet transfer, is decoupled from the control plane responsible for decision-making.

SDN allows the administrator to use less expensive equipment and to strengthen his control of network traffic flows with the use of the OpenFlow protocol, which was initially used to program the control plane of network equipment.

In this memoir, we evaluated the performance of two SDN controllers, these controllers are POX and RYU, and we also realized a Firewall security solution with the most efficient controller in an SDN environment.
Keywords: SDN, controllers, OpenFlow, Pox, Ryu, Firewall.

ملخص

أدى النمو الهائل لمستخدمي الشبكات ومتطلبات الاتصالات الخاصة بهم إلى زيادة كبيرة في استهلاك الطاقة في البنية التحتية للشبكة. ظهر مؤخرا نموذج جديد للشبكات يسمى الشبكات المعرفة بالبرمجيات (اس دي ان) , حيث يتم فصل مستوى البيانات ، المسؤول عن نقل الحزم ، عن مستوى التحكم المسؤول عن صنع القرار.

تسمح اس دي ان للمسؤول باستخدام معدات أقل تكلفة وتعزيز سيطرته على تدفقات حركة مرور الشبكة باستخدام البروتوكول Openflow لبرمجة مستوى التحكم في معدات الشبكة.

في هذه المذكرة، قمنا بتقييم أداء جهازي تحكم اس دي ان، وهما POX و RYU، وأدركنا أيضًا حل أمان جدار الحماية باستخدام وحدة التحكم الأكثر كفاءة في بيئة اس دي ان.

الكلمات المفتاحية : اس دي ان, اوبن-فلو, بوكس, ريو, وحدة التحكم,جدار الحماية.